Conquering the Mobile Device Jungle: Towards a Taxonomy for App-enabled Devices

Christoph Rieger[1] and Tim A. Majchrzak[2]
[1]*ERCIS, University of Münster, Münster, Germany*
[2]*ERCIS, University of Agder, Kristiansand, Norway*

Keywords: App, Mobile App, Taxonomy, Categorization, Smart Devices, Wearable, Smartphone, Tablet.

Abstract: Applications for mobile devices (apps) have created an ecosystem that facilitated a trend towards task-oriented, interoperable software. Following smartphones and tablets, many further kinds of devices became (and still become) app-enabled. Examples for this trend are smart TVs and cars. Additionally, new types of devices have appeared, such as Wearables. App-enabled devices typically share some characteristics, and many ways exist to develop for them. So far for smartphones and tablets alone, issues such as device fragmentation are discussed and technology for cross-platform development is scrutinized. Increasingly, app-enabled devices appear to be a jungle: It becomes harder to keep the overview, to distinguish and categorize devices, and to investigate similarities and differences. We, thus, set out with this position paper to close this gap. In our view, a taxonomy for app-enabled devices is required. This paper presents the first steps towards this taxonomy and thereby invites for discussion.

1 INTRODUCTION

The continuous growth of the mobile device market (Statista Inc., 2016) and the recent emergence of devices such as smartwatches (Chuah et al., 2016) and connected vehicles (Coppola and Morisio, 2016) has attracted much attention from academia and industry. In the past decade, particularly the app ecosystem facilitated a trend towards task-oriented, interoperable software, arguably started with the advent of Apple's iPhone in 2007 (Apple Inc., 2007) and the App Store in 2008 (Apple Inc., 2008). For *traditional* mobile devices (i.e. smartphones and tablets), the competition has yielded two major platforms (Android and iOS). Several approaches for cross-platform development have been proposed to avoid the costly re-development of the same app for different platforms (cf., e.g., Heitkötter, Hanschke, & Majchrzak, 2013; El-Kassas, Abdullah, Yousef, & Wahba, 2015).

Technological development has continued and many new device types have emerged. Most of them fall under the umbrella term *mobile devices* and are more-or-less *app-enabled*. The latter denotes that it typically are apps that make such devices particularly useful and that extend the possibilities they offer. However, they differ greatly in intended use, capabilities, input possibilities, computational power, and versatility, to name just a few aspects. Smart devices such as smart watches and smart TVs are most prominent in the realm of consumer devices but plenty of possibilities exist. Lines towards sensor-driven devices for the Internet of Things (IoT) are often blurred and it is not always clear how to properly categorize a device. This makes it hard to discuss, or, actually, to even correctly name them. Being usually app-enabled, these devices provide new opportunities for intelligent and context-adaptive software but at the same time pose technical challenges regarding the development for new platforms and regarding heterogeneous hardware features. Moreover, app-enablement does not necessarily bring compatibility and portability whereas running the same app on a variety of devices is normally desirable.

While a plethora of case studies and contributions for individual device types such as smartphones and tablets can be found in the scientific literature (e.g., (Heitkötter et al., 2013; Jones and Jia, 2014; Chauhan et al., 2014; Busold et al., 2015; Dageförde et al., 2016)), a comprehensive study of the general field of app-enabled devices is missing. This paper sets out to close this gap by contributing a taxonomy for app-enabled consumer devices. While we have put much effort into literature work (cf. the next section), the useful literature base is small. Therefore, we propose research-in-progress, arguing for our ideas to stimulate

Rieger, C. and Majchrzak, T.
Conquering the Mobile Device Jungle: Towards a Taxonomy for App-enabled Devices.
DOI: 10.5220/0006353003320339
In *Proceedings of the 13th International Conference on Web Information Systems and Technologies (WEBIST 2017)*, pages 332-339
ISBN: 978-989-758-246-2

work on this topic. The eventual aim is the discussion on a conference scale for extension into a state-of-the art paper, possibly further leading to a kind of standard.

The structure of this paper is as follows: After discussing related work in Section 2, our proposal for a taxonomy is presented in Section 3. Section 4 discusses the taxonomy with regard to its current and future applicability before we conclude in Section 5.

2 RELATED WORK

For the topic of our paper on the one hand a plethora of related work exists, on the other hand hardly any closely-related approaches can be cited. Particularly since Apple's iPhone initiated the growth of the smartphone device class, many papers have been published on the *modern* notion of mobile computing, centring around devices that are propelled by apps. However, even overview papers typically focus on one category of devices. For example, Jesdabodi and Maalej (2015) classify apps by usage states but limit themselves to smartphones. Moreover, the scientific literature so far has only rudimentarily captured the latest developments in device development. Chauhan, Seneviratne, Kaafar, Mahanti, and Seneviratne (2016), for instance, provide an overview of smartwatch app markets with focus on the type of apps as well as privacy risks through third party trackers.

To make sure that we do not miss an existing taxonomy (or similar work), we did an extensive literature search. We focus on work from 2012 or later, where the first broader range of smart watches such as the Pebble had already been presented. Together with the increasing variety in devices, new operating systems have appeared since then. Examples are Android Wear and watchOS, which focus on wearable devices (Google Inc., 2016; Apple Inc., 2016) as well as webOS and Tizen, which address a wider range of smart devices (LG Electronics, 2016; The Linux Foundation, 2016).

We deliberately excluded the keywords *application* and *system*. The first yielded many results that were not applicable since the term was mostly used to mean *utilization* of something. The latter had originally been used to describe e.g. cyber-physical systems but now proved to be too generic. Also, the medical area was excluded as these papers focus on apps for therapeutic purposes and do not contribute to the question of app-enabled devices. We thus used the following search string in the Scopus database:

```
TITLE-ABS-KEY(
(app-enabled OR app OR app-based)
AND
(mobile OR smart OR intelligent OR portable)
AND
(device OR vehicle OR ”cyber-physical system”
OR CPS OR gadget)
AND
(classification OR categorization OR overview
OR comparison OR review OR survey OR frame-
work OR model OR landscape OR ”status quo”
OR taxonomy))
AND PUBYEAR AFT 2011
AND ( EXCLUDE ( SUBJAREA , ”MEDI” ) )
```

A search on 07-12-2016 yielded 998 results. Of these, not a single paper provided an approach for classification, let alone a complete taxonomy. Only one paper (Koren and Klamma, 2016) went beyond a perspective on “classical” mobile devices and considered heterogeneous device types. To complicate matters, papers mention that there are other *smart* devices than smartphones but do not go into detail.

In summary, the result set reveals no closely related work to which we can limit ourselves to. However, we can draw from a myriad of sources that tackle *some* aspects that are relevant for a taxonomy of app-enabled devices. This finding aligns with the motivation for our paper. Obviously, other authors struggled with putting different device categories into context because no proper framing exists.

Although not necessarily focused on multiple device categories, work on cross-platform app development is conceptually related. Usually, cross-platform development exclusively targets traditional mobile devices such as smartphones and tablets, e.g. “the diversity in smart-devices (i.e. smartphones and tablets) and in their hardware features; such as screen-resolution, processing power, etc.” (Humayoun et al., 2013). However, considering the differences in platforms, versions, and also at least partly in the hardware is similar to considering a different type of device. In fact, the difference in screen size between some Wearables and smartphones with small screens is less profound than between the same smartphones and tablets. Therefore, comparisons that target cross-platform app development have paved the way towards this paper. This particularly applies to such works that include an in-depth discussion of criteria, such as by Heitkötter et al. (2013), Sommer and Krusche (2013), Dalmasso, Datta, Bonnet, and Nikaein (2013), and Rieger and Majchrzak (2016).

A part of the difficulty with related work is the term *app-enabled* by itself. While it is often said that devices are enabled by apps, or that apps facilitate their functionality, it is usually not explained what this exactly means. The typical usage that we also follow is to denote an app-enabled device as one that by its hardware and basic software (such as the operation system or *platform*) alone provides far less versatility than it is able to offer in combination with additional applications. Such apps are not (all) pre-installed and predominantly provided by third party developers unrelated to the hardware vendor or platform manufacturer; moreover, the possibilities provided by apps typically increase over time *after* a device has been introduced. While this still is no profound definition, it provides a demarcation for the time being. In particular, it rules out pure Internet-of-Things devices as well as computational equipment that only is occasionally firmware-updated or that is not built for regular interaction with human users.

3 TAXONOMY OF APP-ENABLED DEVICES

Categorizing app-enabled devices is difficult due to the variety of possible hardware features *across* all types of devices and the heterogeneity of device capabilities *within* each class. Any simple solution is prone to not sufficiently discriminate. For example, processing power does not differ a lot between smartphones and tablets anymore, and microphones are no distinguishing feature for voice-controlled devices. In addition, the fast-paced technological progress manifests as a constant stream of new devices, partly rendering previous devices obsolete. Moreover, device types converge, illustrated e.g. by the *phablet* phenomenon (devices that fall in between smartphones and tablets). Mobility in the strict sense even is no exclusive feature; smart TVs for example are not really mobile. Cars with smart entertainment systems or even self-driving features might be app-enabled, but it can be disputed whether the whole car is the *device* and thereby the device is actually mobile by itself. As a result, a taxonomy of app-enabled devices mandates a more open categorization along several dimensions, allowing for partial overlaps and future additions. In the following, we present steps towards such a taxonomy. First, the three chosen dimensions are discussed as static structure for classification. Subsequently, current and foreseeable future device classes are positioned according to this matrix.

3.1 Dimensions of the Taxonomy

This work positions app-enabled devices with regard to the three dimensions *media richness of inputs*, *media richness of outputs*, and the *degree of mobility*. Instead of enumerating concrete technologies that are available today or in future, each dimension should rather be regarded as continuously increasing intensity and flexibility of the particular capability, with several exemplary cornerstones depicted in the following. This approach not only provides the highest degree of objectivity but also should keep the taxonomy flexible enough to capture future developments without actually changing the dimensions.

Media richness of inputs describes the characteristic user input interface for the respective device class.

- *None* refers to fully automated data input through sensors.[1]
- *Buttons*, including switches and dials, are (physically) located at the device itself and provide limited input capabilities.
- *Remote controls*, including also joysticks and gamepads, refer to dedicated devices that are tethered or wirelessly connected to the app-enabled device.
- *Keyboards* are also dedicated devices to control the target devices, but with more flexible input capabilities due to a variety of keys. Input still is discrete.
- *Pointing devices* refer to all dedicated devices to freely navigate and manipulate the (mostly graphical) user interface, for example mouse, stylus, and graphic tablet. While these devices technically still provide discrete input, the perception of input is continuous.
- *Touch* adds advanced input capabilities on the device itself, allowing for more complex interactions such as swipe and multi-touch gestures.
- *Voice*-based devices are not bound to tangible input units but can be controlled without haptic contact.
- *Gestures* allow for a hands-free visual user interaction with the device, for example using gloves or motion sensing.
- *Neural interfaces* are the richest form of user inputs in the future by directly tapping into the brain or nervous system of the human operator.[2]

As second dimension, *media richness of outputs* describes the main output mechanisms for the respective device class.

[1] Strictly, most if not all input is done via sensors, but *none* at this point denotes no manual activity by a user.

[2] Since the possibilities of neural interfaces are yet very limited, future developments *might* mandate splitting up this category into different kinds of neural interfaces.

- *None* refers to no user-oriented communication by the device itself. This applies to cyber-physical actuators with direct manipulation of real-world objects (e.g. switching on light). This also includes pass-through mechanisms that in general or in some situations do not produce a tangible output of their own but pass it through to a connected managing device (e.g., a smartphone) which retrieves information and handles user output.
- *Screen* output is a major form of user communication found in app-enabled devices. Although a clear subdivision is not possible, several classes are typical, ranging from tiny screen displays (<3") to small screens such as for smartphones (<6"), medium screens for handheld devices (<11"), large screens (≤ 20"), and usually permanently installed huge screens >20".
- *Voice*-based output refers to the first type of disembodied device output that communicates with the user without physical contact.
- *Projection* extends the disembodiment with visual output to a device-external location, including augmented reality applications and hologram representations.
- *Neural interfaces* connect directly to the user in order to a achieve a tightly coupled human-computer interaction.

Finally, the combination of input and output characteristics ignores different application areas of the respective device class. For example, intelligent switches and drones for aerial photography can both be remotely controlled and have no direct output, but can hardly be grouped as being in the same device class. Therefore, the *degree of mobility* describes the usage characteristics as third dimension.

- *Stationary* devices are permanently installed and have no mobile characteristics during use.
- *Mobile* devices can be carried to the place of use.
- *Wearable* devices are designed for a more extensive usage and availability through the physical contact with the user. In contrast to "mobile", transporting the device is implicit and usually hands-free.
- *Self-moving* devices provide the capability to move themselves (controlled by the user). Ultimately, autonomous devices represent the richest form of mobility for app-enabled devices.

3.2 Categorizing the Device Landscape

The proposed dimensions allow for an initial categorization of the device landscape. Figures 1 to 3 (page 5) visualize the three-dimensional categorization of different device classes using the respective two-dimensional projections for better readability.

As depicted in Figure 1, many devices classes can be assigned to distinct positions in the two-dimensional space of input/ output media richness. However, it should be noted that the ellipses represent (current) major interaction mechanisms within the device class. For example, *smartphones* also have a few physical buttons but are usually operated by touch input. Individual devices may also deviate from the presented position, for instance specialized or experimental devices that do not (yet?) constitute a distinct class of devices. Additionally, not all devices falling into a device class must necessarily implement all possibilities of that class. Therefore, ellipses are a well-suited representation as opposed to, e.g., the maximum value for the respective devices.

The chosen level of abstraction implies that the taxonomy *dimensions* are intended to be rather static. Instead of chasing the actual technological development to reflect the latest emergence of devices, only seldom and slow changes are necessary to keep them up to date. Nevertheless, the categorization of *classes* is more dynamic and will need to be regularly checked for continued relevance. Moreover, classes might need to be split or at least be adapted regarding their placement on the dimensions' continuum when new possibilities arise. For this reason, and also for the brevity of a position paper such as this, we do not provide detailed definitions for each class. Rather, we explain them exemplarily and rely on the general understanding of the well-known classes (such as smartphones).

Figure 1 reveals differences in the specificity (i.e., represented size) of the device classes. Some of them fill specific spots in the diagram, either due to technical restrictions (*smart TVs* evolve traditional remote-controlled TVs with large screens) or special purposes (*smart glasses* enable hands-free interaction and visualization). Less specific device classes exist for two reasons. On the one hand, terms such as *smart home* comprise every technology that relates to a specific domain, subsuming very heterogeneous devices. On the other hand, *underspecified* device classes such as *implants* and *smart personal agents* are presented as they are due to their novelty; there are few devices on the market and a high level of uncertainty must be ascertained regarding future hardware characteristics and interaction patterns. We expect to be able to draw a more concrete picture with a future version of the taxonomy, both based on the discussion of this paper and the meanwhile ongoing technological progress.

Differences in the device classes can also be explained with regard to *media richness theory* (MRT).

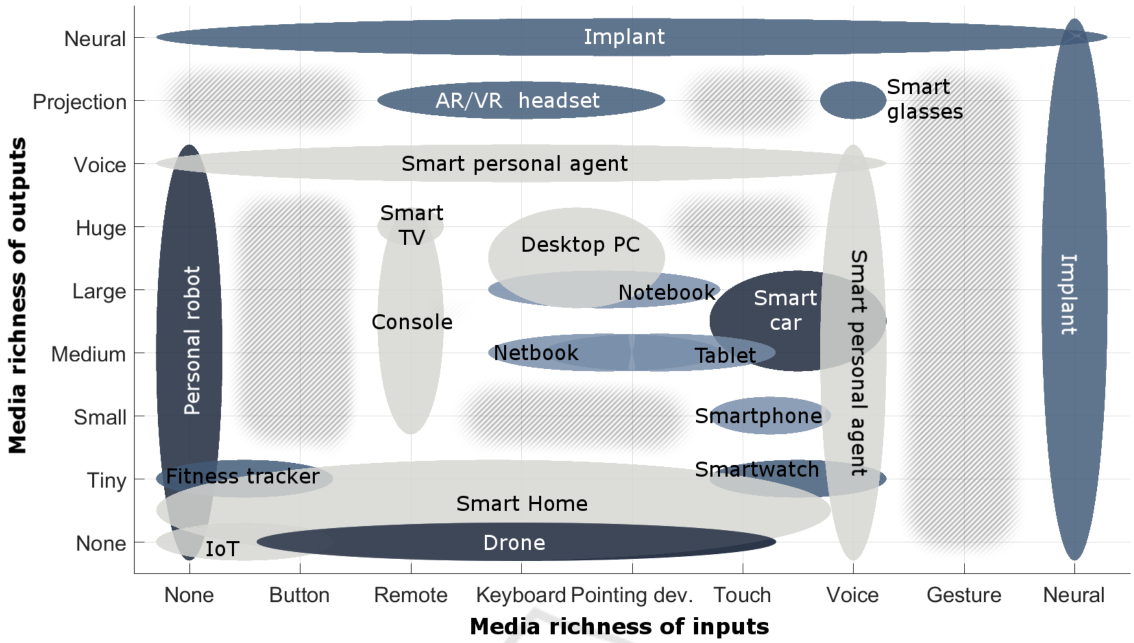

Figure 1: Matrix of Input and Output Dimensions.

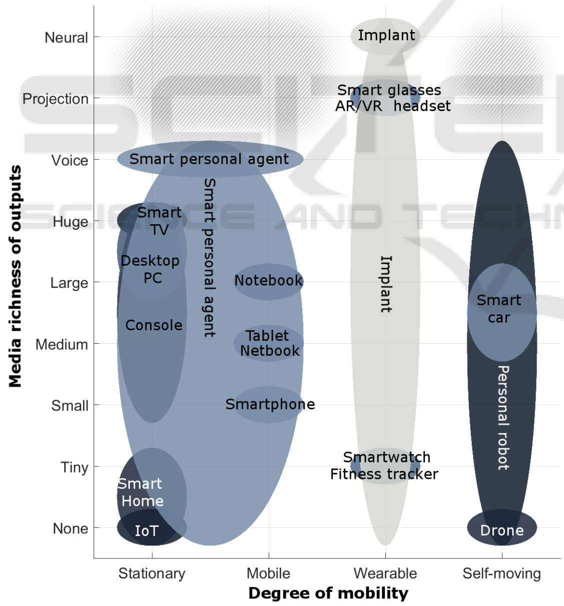

Figure 2: Matrix of Output and Mobility Dimensions.

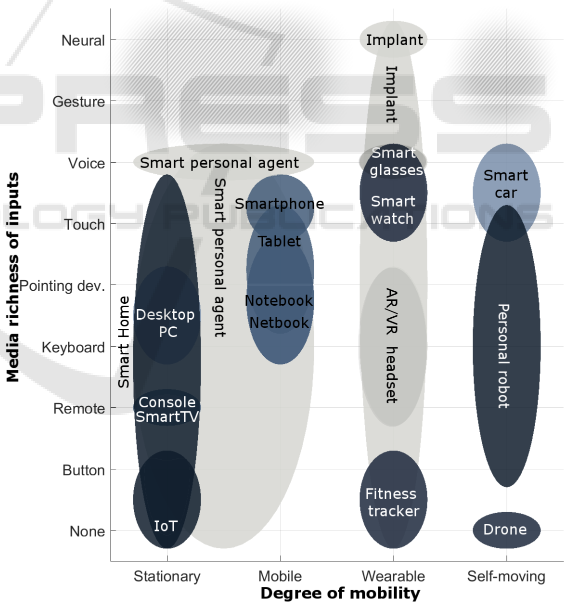

Figure 3: Matrix of Input and Mobility Dimensions.

MRT describes a corridor of effective communication with matching levels of message ambiguity and media richness (Daft et al., 1987). When applying this idea to the input and output characteristics of app-enabled devices, similar observations can be made. For example, *IoT* devices have only rudimentary direct user input possibilities but also give not much feedback in return. Notebooks allow for medium levels of input richness through keyboard and mouse input, with large screens as more flexible output capabilities. Furthermore, *smart glasses* directly embed their output into the real world by projection. Consequently, their voice-based input is equally rich in order to handle complex interactions with the user.

This correlation also partly explains why there are areas in Figure 1 with no assigned device class. Rich forms of user input such as gestures overcomplicate interactions for devices that have just small screens. On the other extreme, devices with barely a few buttons do not provide sufficiently flexible input capabilities to manipulate large screens. Of course, empty spaces in the taxonomy might also be caused by a lack of use cases so far. Thus, they might actually be filled by future devices, or existing classes might "stretch" into these areas. In general, with the evolution and differentiation of input media, existing device classes might extend towards further areas or even converge. E.g., consider convertibles as hybrid devices between keyboard-based *netbooks* and touch-optimized *tablets*.

Figure 2 depicts the combination of output media richness and mobility. Unsurprisingly, a general tendency towards large screen output for stationary devices can be observed. With increasing mobility, screen sizes tend to diminish, from small screens on *smartphones* to very limited *fitness tracker* screens and screen-less *drones*. Beyond mobile devices, output capabilities become more flexible, potentially caused by specific domains of application, or their novelty with insufficient time to establish wide-spread interaction patterns. Some recently emerged device classes explore intangible output capabilities, for instance *augmented / virtual reality (AR/VR) headsets*. Others, especially autonomously moving devices such as *smart cars* and *personal robots*, are driven by the increased availability of sensor technology and not restricted to particular output capabilities.

Finally, Figure 3 visualizes the relationship between input media richness and mobility. Usually, an increasing degree of mobility entails less physical input mechanisms with dedicated buttons and keys. This might be attributed to practicability reasons, for example using voice commands is easier for wearable *smart glasses* than requiring dedicated input devices. In addition, smarter devices are usually more complex with regard to their output, and equally sophisticated input capabilities are necessary to match this level as explained by media richness theory. *Consoles*, for instance, provide basic navigation functionalities. *Desktop personal computers* and *notebooks* can be equipped with intelligent software such that keyboard and mouse are helpful means for interaction, and *smart personal agents* integrate advanced interpretation mechanisms that allow for voice-based communication in everyday situations.

4 DISCUSSION

Due to the rapid proliferation of the field and the initial character of this work, it is likely that amendments will need to be made. Moreover, we will need to keep updating the taxonomy once it has been acknowledged by the scientific community. In addition, a taxonomy should also be appealing for the use by practitioners, particularly in a field like Mobile Computing where scientific research and technological progress go hand-in-hand. Therefore, this section presents ideas for discussion that go beyond Section 3.

4.1 Alternative Categorization Schemes

Devices can be categorized according to other device features. Not all are compatible with our taxonomy, nevertheless we deem several of them noteworthy.

Simple schemes such as a categorization by hardware feature (e.g., camera, computing power, touch screen) or usage (e.g., business, entertainment, sports, or communication use) fail to provide clear criteria for a taxonomy. In particular, a fast adaptation and convergence of available technologies could be observed in the past years. For example, so-called *phablets* blur the lines between smartphones and tablets, and gyroscope sensors have found wide-spread adoption in a variety of devices.

Matrix-based categorizations allow for a better juxtaposition on two dimensions, for instance regarding the input and output characteristics of app-enabled devices. However, the heterogeneity of devices within a device class provides insufficient discriminating power. For example, medium-sized and touch-based screens are usual interfaces both for tablets and smart cars. Similarly, distinguishing between apps for embedded or stand-alone devices is not always possible due to different types of device integrations within a device category (cf. e.g. Coppola and Morisio (2016) for smart cars).

Therefore, the third dimension chosen for our taxonomy adds the degree of mobility to distinguish between similar device hardware in different usage contexts. Other potential approaches for categorizing devices include the degree of integration, automation, or intelligence attainable or provided by the device. This reaches from simple input / output devices with limited app interaction (such as fitness trackers), to interoperable software (such as smartphones), highly cross-linked and automated devices (in the IoT or smart home field), and finally to intelligent machines. While we deem it reasonable to discuss such an optional fourth dimension, we do not think the taxonomy would gain more discriminatory power.

4.2 Further Development

Firstly, future discussion needs to include the demarcation of devices to be included. As argued earlier, mobility is not necessarily the proper boundary. App-enablement has proven to be feasible, yet we will need to find (or provide) a profound definition for it.

Secondly, it needs to be determined how the taxonomy can be kept up to date. In many other cases, taxonomies have proven to be either too detailed and thus requiring constant adjustments, or too little detailed and thus lacking discriminatory power. Due to a restriction to three orthogonal dimensions and clearly distinguishable values in each of it, we are optimistic that the taxonomy will be future-proof. Nevertheless, proper ways of deciding when adaptations are needed and what developments can be reflected without changes need to be defined. As part of this, we will need to scrutinize how to handle the differences in precision regarding categories. For example, it is very well understood what a smartphone is; smart homes, and to an even higher degree neural devices are (yet) diffuse with a lack of devices and/or applications to characterize them.

Thirdly, we so far have limited ourselves to consumer devices. This includes many devices that are *also* used for professional purposes, but arguably not all. Beyond that, some specialised devices are (so far) solely used for professional means. Examples can be found in industry, particularly in logistics. However, some of these might simply be subsumed by consumer devices. It could be said that e.g. the devices used by parcel couriers are very similar to smartphones, despite the difference in form and the absence of a general purpose utilization. The same applies to special devices from areas such as healthcare or crisis prevention and response. While such devices typically have specific capabilities (such as error-tolerance), on an abstract level they again are very similar to general purpose hardware. Thus, an updated taxonomy could try to include non-consumer devices. However, due to the complexity that arises particularly with devices that are so specialised that information on them is scarce, we deem the current limitation justified.

Fourthly, it should be scrutinized how the taxonomy can be provided in a form that is useful both for researchers and for practitioners. Most scientists know taxonomies for research topics enforced by publication outlets. Quite often these feel more like a “try to fit somewhere” game, particularly if a paper tackles a contemporary topic and the taxonomy provides little flexibility. If we want our taxonomy to be helpful for researchers, and – probably even hard to achieve – employed by practitioners, it needs to be easy to use yet powerful. Achieving this will be very valuable, as can e.g. be seen for cross-platform development, where new approaches can be clearly categorized by their characteristics.

The above discussion points have also shown the limitations of our work. Besides the issues that need to be worked on, an eventual verification of the taxonomy is mandated. For now, many tasks remain but a first – we deem noteworthy – step has been done. Further steps are sketched in the next Section.

5 CONCLUSION AND OUTLOOK

In this paper we have presented a taxonomy for app-enabled devices – it is the first such work. Based on three dimensions that take into account the input and output characteristics, we have built the taxonomy. Categorizing the device landscape and plotting the results into figures illustrates the discriminatory power of our taxonomy. However, as the first comprehensive work on the topic, we seek to amend and refine it. For now, it is put up for discussion. Nonetheless, due to the soundness of the dimensions and their alignment with availably theory, we are optimistic that this position paper provides a profound step towards a systematic overview of app-enabled devices.

Our future work instantaneously follows this precondition. We seek to discuss the taxonomy as part of a conference presentation and, subsequently, with interested colleagues. Moreover, we will also seek to have practitioners scrutinize it. The next step will be to provide an amended version of the taxonomy, along with an extended discussion of device classes. Eventually, empirical verification will be necessary.

The outlook is determined by an aspiration for our work. It should not become “yet another computer science taxonomy”. Rather, it should prove useful in allowing

- authors to more clearly express what kind of device(s) they are referring to,
- researchers and practitioners to gain more discriminatory power when speaking about modern mobile computing devices, and
- the general public a more straightforward way of understanding similarities and differences between devices, both technically and tangibly.

REFERENCES

Apple Inc. (2007). *Apple reinvents the phone with iPhone.* Retrieved 11.01.2017, from http://www.apple.com/pr/library/2007/01/09Apple-Reinvents-the-Phone-with-iPhone.html

Apple Inc. (2008). *iPhone app store downloads top 10 million in first weekend.* Retrieved 11.01.2017, from http:// www.apple.com/pr/library/2008/07/14iPhone-App-Store-Downloads-Top-10-Million-in-First-Weekend.html

Apple Inc. (2016). *watchOS.* Retrieved 12.01.2017, from www.apple.com/watchos/

Busold, C., Heuser, S., Rios, J., Sadeghi, A. R., and Asokan, N. (2015). Smart and secure cross-device apps for the internet of advanced things. *LNCS*, 8975:272–290. doi: 10.1007/978-3-662-47854-7_17

Chauhan, J., Mahanti, A., and Kaafar, M. A. (2014). Towards the era of wearable computing? In *Pro. 2014 CoNEXT on Student Workshop*, CoNEXT Student Workshop '14, pages 24–25. ACM. doi: 10.1145/2680821.2680833

Chauhan, J., Seneviratne, S., Kaafar, M. A., Mahanti, A., and Seneviratne, A. (2016). Characterization of early smartwatch apps. *2016 IEEE Int. Conf. on PerCom Workshops 2016*. doi: 10.1109/PERCOMW.2016.7457170

Chuah, S. H.-W., Rauschnabel, P. A., Krey, N., Nguyen, B., Ramayah, T., and Lade, S. (2016). Wearable technologies: The role of usefulness and visibility in smartwatch adoption. *Computers in Human Behavior*, 65:276–284. doi: 10.1016/j.chb.2016.07.047

Coppola, R. and Morisio, M. (2016). Connected car: Technologies, issues, future trends. *ACM Comput. Surv.*, 49(3):46:1. doi: 10.1145/2971482

Daft, R. L., Lengel, R. H., and Trevino, L. K. (1987). Message equivocality, media selection, and manager performance: Implications for information systems. *MIS quarterly*, 11(3):355. doi: 10.2307/248682

Dageförde, J. C., Reischmann, T., Majchrzak, T. A., and Ernsting, J. (2016). Generating app product lines in a model-driven cross-platform development approach. In *49th HICSS*, pages 5803–5812. doi: 10.1109/HICSS.2016.718

Dalmasso, I., Datta, S. K., Bonnet, C., and Nikaein, N. (2013). Survey, comparison and evaluation of cross platform mobile application development tools. In *Proc. 9th IWCMC*. doi: 10.1109/IWCMC.2013.6583580

El-Kassas, W. S., Abdullah, B. A., Yousef, A. H., and Wahba, A. M. (2015). Taxonomy of cross-platform mobile applications development approaches. *Ain Shams Engineering Journal*.

Google Inc. (2016). Android wear. Retrieved 12.01.2017, from https://android.com/wear

Heitkötter, H., Hanschke, S., and Majchrzak, T. A. (2013). Evaluating Cross-Platform Development Approaches for Mobile Applications. In Cordeiro, J. and Krempels, K., editors, *Revised Selected Papers WEBIST 2012*, volume 140 of *Lecture Notes in Business Information Processing (LNBIP)*, pages 120–138. Springer.

Heitkötter, H., Majchrzak, T. A., and Kuchen, H. (2013). Cross-platform model-driven development of mobile applications with MD2. In Shin, S. Y. and Maldonado, J. C., editors, *Proc. 28th ACM SAC*, SAC '13, pages 526–533. ACM. doi: 10.1145/2480362.2480464

Humayoun, S. R., Ehrhart, S., and Ebert, A. (2013). Developing mobile apps using cross-platform frameworks: A case study. In Kurosu, M., editor, *Proc. 15th Int. Conf. HCI International, Part I*, pages 371–380. Springer. doi: 10.1007/978-3-642-39232-0 41

Jesdabodi, C. and Maalej, W. (2015). Understanding usage states on mobile devices. In *Proc. 2015 ACM Int. Joint Conf. on Pervasive and Ubiquitous Computing*, UbiComp '15, pages 1221–1225. ACM. doi: 10.1145/2750858.2805837

Jones, C. and Jia, X. (2014). The AXIOM model framework: Transforming requirements to native code for cross-platform mobile applications. In Ferreira Pires, L., editor, *2nd Int. Conf. on Model-Driven Engineering and Software Dev.* IEEE.

Koren, I. and Klamma, R. (2016). The direwolf inside you: End user development for heterogeneous web of things appliances. In Bozzon, A., Cudre-Maroux, P., and Pautasso, C., editors, *Proc. 16th Int, Conf, ICWE 2016*, pages 484–491. Springer. doi: 10.1007/978-3-319-38791-8 35

LG Electronics (2016). WebOS for LG Smart TVs. Retrieved 31.05.2016, from http://www.lg.com/uk/smarttv/webos

Rieger, C. and Majchrzak, T. A. (2016). Weighted evaluation framework for cross-platform app development approaches. In Wrycza, S., editor, *Proc. 9th SIGSAND/PLAIS EuroSymposium*, pages 18–39. Springer. doi: 10.1007/978-3-319-46642-2_2

Sommer, A. and Krusche, S. (2013). Evaluation of cross-platform frameworks for mobile applications. *Lecture Notes in Informatics*, P-215:363–376.

Statista Inc. (2016). Global smartphone shipments forecast from 2010 to 2020. Retrieved 11.01.2017, from https://www.statista.com/statistics/263441/

The Linux Foundation. (2016). *Tizen. Retrieved 12.01.2017, from https://www.tizen.org*