A Taxonomy for App-Enabled Devices: Mastering the Mobile Device Jungle

Christoph Rieger[1(✉)] and Tim A. Majchrzak[2]

[1] ERCIS, University of Münster, Münster, Germany
christoph.rieger@ercis.de
[2] ERCIS, University of Agder, Kristiansand, Norway
tima@ercis.de

Abstract. While the term application is known for a long time, what we now refer to as mobile apps has facilitated task-oriented, interoperable software. The term was initially only used for smartphones and tablets, but desktop software now is also referred to as apps. More important than the wording, however, is the trend towards app-enablement of many further kinds of devices such as smart TVs and wearables. App-enabled devices usually share some characteristics and developing apps is often similar. However, many complexities must be mastered: Device fragmentation and cross-platform app development already are challenging when only considering smartphones. When trying to grasp the field as a whole, app-enabled devices appear as a jungle: it becomes increasingly hard to get an overview. Devices might not be easy to categorize let alone to compare. Investigating similarities and differences is not straightforward, as the outer appearance might be deceiving, and technological peculiarities are often complex in nature. This article aims at mastering the jungle. For this purpose, we propose a taxonomy for app-enabled devices. It provides clear terms and facilitates precision when discussing devices. Besides presenting the taxonomy and the rationale behind it, this article invites for discussion.

Keywords: App · Mobile app · Taxonomy · Categorization
Smart devices · Wearable · Smartphone · Tablet

1 Introduction

The continuous growth of the mobile device market [2] and the recent emergence of devices such as smart watches [3] and connected vehicles [4] has attracted much attention from academia and industry. In the past decade, particularly

This article greatly extends the short paper [1] presented at WEBIST 2017. It has been updated to reflect the latest developments, includes new content based on additional work as well as on the discussions at the conference, and has been amended with a comprehensive discussion. Please note that verbatim content from the short paper is not explicitly highlighted but for figures and tables already included there.

T. A. Majchrzak et al. (Eds.): WEBIST 2017, LNBIP 322, pp. 202–220, 2018.
https://doi.org/10.1007/978-3-319-93527-0_10

the app ecosystem facilitated a trend towards task-oriented, interoperable software, arguably started with the advent of Apple's iPhone in 2007 [5] and the App Store in 2008 [6]. For *traditional* mobile devices (i.e. smartphones and tablets), the competition has yielded two major platforms (Android and iOS). However, whether these two will prevail can hardly be estimated, yet. Moreover, developing for such devices in a unified way is still not possible in all cases and with ease (cf. also [7]). 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., [8,9]).

Technological development has continued in the meantime and many new device types have emerged. Most of them fall under the umbrella term *mobile devices* and are concerned by the research field of *mobile computing*. Typically, they are more-or-less *app-enabled*. While app enablement is no fixed or even defined term (to the best of our knowledge), it can be understood as follows:

> An app-enabled device provides hardware that allows it to be used for multiple (typically many) purposes and in changing contexts – possibly even unforeseen by the device manufacturer – while the actual versatility of the device is achieved through means of extensible software that comes in small, interchangeable pieces which are usually provided by third parties.

Thus, it typically are apps that make such devices particularly useful and that extend the possibilities they offer. However, mobile devices that follow our rough definition differ greatly in intended use, capabilities, input possibilities, computational power, and versatility, to name just a few aspects. In early visions of a world connected by ubiquitous mobile devices, these were only thought of as tabs, pads, and boards [10]. So-called "smart devices" such as smart watches and smart TVs are most prominent in the realm of consumer devices and exhibit double-digit sales growths over the past years [11,12], but plenty of possibilities exist with regard to the physical embodiment of virtual assistants. Furthermore, hardware in professionally contexts can be surprisingly similar to consumer-hardware; apps can make them seem even more akin. 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 [13]. This makes it hard to discuss, or, actually, to even correctly name them. The resulting blurriness makes it hard to delimit research and practical work. Much worse, when speaking and writing about mobile devices, the level of precision is often not as high as it is when well-known concepts are discussed. While this is normal for emerging fields, it is particularly noticeable for work on mobile computing. To our observation, there are only slow improvements.

We believe that more precision in speaking and writing will eventually also be beneficial for research on mobile devices and their app-enablement. These devices provide a plethora of new opportunities for intelligent and context-adaptive software. At the same time, they pose technical challenges regarding the development for new platforms and regarding heterogeneous hardware features. Interestingly, these challenges can be quite similar despite seemingly very different devices, as they can be completely different despite originating from the same kind of device. Moreover, app-enablement does not necessarily bring compatibility and portability. Naturally, running the same app on a variety of devices is normally desirable. If we still rely on cross-platform approaches and search for a development unifier merely for smartphones and tablets [7], developing for heterogeneous mobile devices is an endeavour far greater in complexity. It would probably be ideal to reach something like a progression in functionality: the same app would function on many devices but respectively provide the highest level of functionality achievable on the given hardware and with the available other software. It must be doubted, however, that such an ideal can be reached as long as we do not even properly know *what* we are talking about.

While a plethora of case studies and contributions for individual device types – mainly focused an smartphones and tablets – can be found in the scientific literature (e.g., [14–18]), a comprehensive study of the general field of app-enabled devices is missing. With our WEBIST position paper [1], we set out to close this gap by contributing a taxonomy for app-enabled consumer devices. This contribution got favourable comments, encouraging us to provide an extension of our work with this article. The taxonomy aims at

– helping authors to clearly express what kind of device(s) they refer to,
– providing researchers and practitioners with more discriminatory power when referring to topics from modern mobile computing, and
– giving the general public a more straightforward understanding of similarities and differences between devices, both technically and tangibly.

Similar as in the prior paper, we have put much effort into literature work (cf. the next section), although the useful literature remains scarce. While the taxonomy has only been slightly updated to reflect the latest developments, we delve deeper into the theoretic dimension and also extend our discussion. Therefore, the work presented in this paper keeps a research-in-progress flavour, since it is impossible to suggest that our taxonomy is in its final state. However, it should be considered sufficiently stable for practical application. Any follow-up work from now on will honour this by either providing downward compatibility and (or alternatively) by explicating changes. We believe that more work will continue to be required; while we of course hope for this article to become a state-of-the reference, it should also stimulate further work. The mid-term goal remains to be a *de-facto* standard.

The remainder of this article is structured as follows. Section 2 takes an updated look at relevant literature in the narrow sense; related work to specific aspects is referenced throughout the paper. Then, our proposal for a taxonomy is presented in Sect. 3. Section 4 discusses the taxonomy with regard to its current and future applicability. Finally, we conclude and give an outlook in Sect. 5.

2 Related Work

If you consider the topic of our article broadly, a plethora of related work exists. Looking at it in more detail, hardly any closely-related approaches can be cited.

This is not really surprising: all papers that deal with apps and app-enabled devices must (at least implicitly) explain *what* they actually deal with. However, no systematic work exists that defines kinds of devices, modes of app-enablement, notions of mobility of devices, and so on.

Particularly since Apple's iPhone founded the *smartphone* device class, which soon saw many devices follow, 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, [19] 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. [20], for instance, provide an overview of smart watch 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 conducted 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 [21,22] as well as webOS and Tizen, which address a wider range of smart devices [23,24]. Additionally, also the app ecosystems have matured, with HTML5 gaining momentum and possible technological unifiers such as progressive web apps (PWAs) [7] emerging.

In our search, 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 framework OR model OR landscape OR “status quo” →
    OR taxonomy)
)
AND PUBYEAR AFT 2011
AND ( EXCLUDE ( SUBJAREA, “MEDI” ))
```

A search on 01-08-2017 yielded 1,268 results. Of these, not a single paper provided an approach for classification, let alone a complete taxonomy. To complicate matters, some papers mention that there are other *smart* devices than smartphones and tablets but do not go into detail. Only four papers went beyond a perspective on "classical" mobile devices: Some authors focus on specific combinations of devices, including Neate et al. [25] who analyse the use case of second screening that combines smart TVs with additional mobile devices and Singh and Buford [26] who describe cross-device team communication apps for desktop, smartphones and wearables. Regarding more generalized approaches, Queirós et al. [27] focus on context-aware apps also suitable for novel mobile devices using the example of an automotive app. Finally, Koren and Klamma [28] considered the integration of heterogeneous Web of Things device types by adopting a middleware approach.

In summary, the result set reveals no closely related work to which we can limit ourselves. 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.

Despite not necessarily focussing 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." [29]. 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 (such as some smart watches) 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 article. This particularly applies to such works that include an in-depth discussion of criteria, such as by [8,30–32].

A part of the difficulty with related work is the term *app-enabled* (or *app-enablement*) 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. But in the simplest devices that make use of computer hardware, software plays an important role; in consequence, merely being capable of running software that fulfills more than basic functionality is not enough to describe the term.

The typical usage that we also follow is to denote an app-enabled device as one that by its hardware and foundational 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. In addition, apps may expose use cases not originally intended or even imagined. 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

In the following, we describe the preconditions of a taxonomy before describing how a device categorization can be tailored. We deem three dimensions to be viable as the static structure for classification. Eventually, we propose a categorization that captures the status quo. It positions current and foreseeable future device classes according to this matrix.

3.1 Basic Considerations

Categorizing app-enabled devices is difficult: there is a wide variety of possible hardware features *across* all types of devices, which even further increases. For example, in the past fingerprint scanners were restricted to few notebooks but today also appear on smartphones because of simplified user authentication and changing security requirements with regard to the device purpose (consumer vs. commercial). If classes are set – such as the widely acknowledged distinction between smartphones and tablets – there still is a heterogeneity of device capabilities *within* each class. For instance, the first smartwatches offered only a few sensors. Current devices have many more sensors, and their characteristics can differ significantly depending on the target sector (such as low-end vs. high-end).

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. 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.

3.2 Dimensions of the Taxonomy

We position 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 *may be* introduced in the future, each dimension should rather be regarded as continuously increasing intensity and variability 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. Thereby, it captures how human users can interact with a device. Additional machine-to-machine communication through the same or distinctive interfaces is *not* considered.

None refers to fully automated data input through sensors.[1]

Pass-through represents the indirect manipulation through data exchange with an external device (which in turn might originate from user input) whose purpose is not solely to provide the user interface for the main device.[2]

Buttons including switches and dials, are (physically) located at the device and provide rather 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. Technically, they merely make use of buttons, switches, dials etc. but provide a richer experience due to being decoupled from the 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. Strictly speaking, within this category simple touch events and several forms of increasingly complex multi-touch gestures can be subdivided.

Voice-based devices are not bound to tangible input surfaces but can be controlled without haptic contact.

Gestures allow for a hands-free user interaction with the device, for example using gloves or motion sensing. Technologically, different solutions are possible, e.g. based on gyroscopes, cameras, and lidars [33,34].

Neural interfaces can be expected to become the richest form of user inputs by directly tapping into the brain or nervous system of the human operator.[3]

As the second dimension, *media richness of outputs* describes the main output mechanisms for the respective device class. Similarly to the input, human users

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

[2] For example, an autonomous device with a companion smartphone app for remote handling can be subsumed under this category.

[3] Since the possibilities of neural interfaces are yet very limited and any work so far is experimental, future developments *might* mandate splitting up this category into different kinds of neural interfaces.

are concerned as the receivers; possible machine-to-machine communication is not relevant for this dimension.

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).

Pass-through includes mechanisms that in general or in some situations do not produce human-directed 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 the prevalent form of user communication found in app-enabled devices. Although a clear subdivision is not possible, several classes are typically observed, ranging from tiny screen displays ($<3''$) to small screens such as for smartphones ($<6''$), medium screens for handheld devices ($<11''$), large screens ($\leq 20''$), and usually permanently installed huge screens $>20''$.

Projection refers to the first type of disembodied device output to a device-external surface without physical contact.

Voice-based output extends the disembodiment with auditive output to communicate with the user without physical contact.

Augmented reality includes virtual reality applications and hologram representations, further increases the richness of device outputs by modifying or fully replacing the perceived reality around the user.

Neural interfaces connect directly to the user in order to a achieve a tightly coupled human-computer interaction.[4]

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. Whereas several studies deal with usage characteristic particular devices such as smartphones (e.g., [35]), to the best of our knowledge no closely related work exists on context-dependent device usage across different mobile devices. Therefore, the *degree of mobility* describes the usage characteristics as the third dimension on a high level. With regard to trends such as ubiquitous computing [36], this dimension also reflects the pervasiveness and integration of mobile devices in everyday activities – from on-demand usage of stationary devices to always connected autonomous assistants.

Stationary devices are permanently installed and have no mobile characteristics during use.

Moveable devices can be carried to the place of use. This includes an "on-the-go" utilization, such as a smartphone being used while walking.

[4] Similarly to the considerations for the input, it will need to be seen whether neural interfaces for output require some form of subdivision. Technology so far is in an early experimental state.

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 often hands-free.

Self-moving devices provide the capability to move themselves (directly or indirectly controlled by the user). Ultimately, autonomous devices represent the richest form of mobility for app-enabled devices.

3.3 Categorizing the Device Landscape

The proposed dimensions allow for an initial categorization of the device landscape. Figures 1, 2 and 3 (pages (9) to (11)) visualize the three-dimensional categorization of different device classes using three two-dimensional projections for better readability. Also, Table 1 summarizes the device classes discussed in this paper.

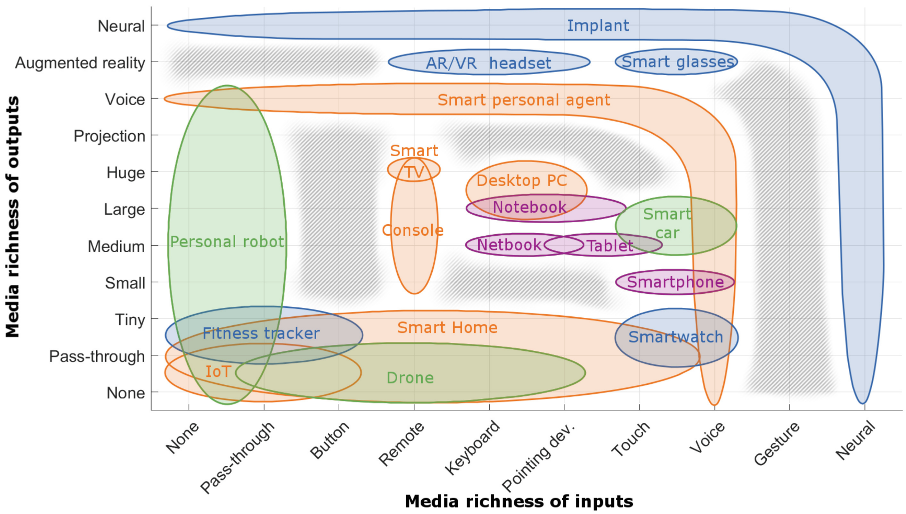

Fig. 1. Matrix of input and output dimensions (adapted from [1]).

As depicted in Fig. 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 classes. For example, *smartphones* also have a few physical buttons but are mainly 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, devices might be extended. For example, through special plugs computer mouses can typically be attached to smartphones. Since this is normally meant for debugging purposes, "pointing devices" would not normally be considered an

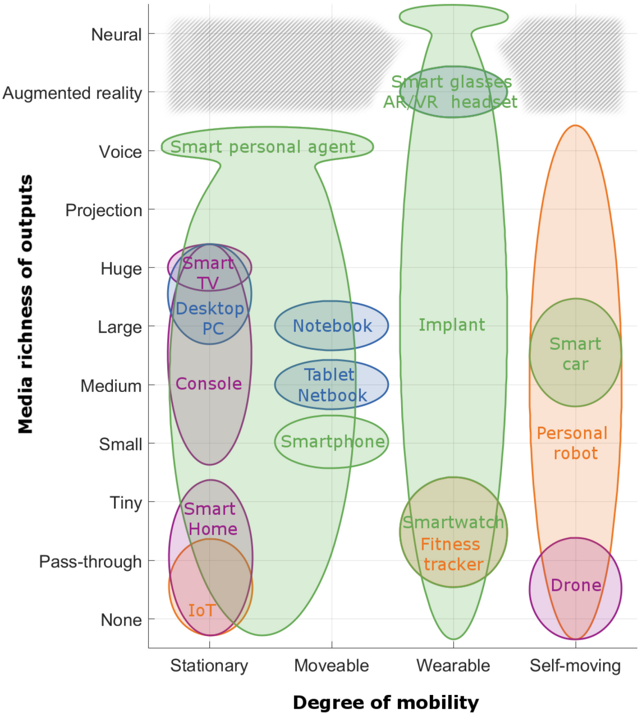

Fig. 2. Matrix of output and mobility dimensions (adapted from [1]).

input for smartphones. Similarly, so-called pico projectors allow image projection from smartphones and tablets. They are no typical mean for output, although this might change in the future.

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. A good approximation would be to consider at least 80% of all devices to match a category, with the lowest and the highest decile being outliers. For example, *convertibles* as hybrid devices between keyboard-based notebooks and touch-oriented tablets are not considered as they still represent a small minority in both categories.

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. Thus, we explain some noteworthy classes exemplarily and rely on the general understanding of the well-known classes (such as smartphones).

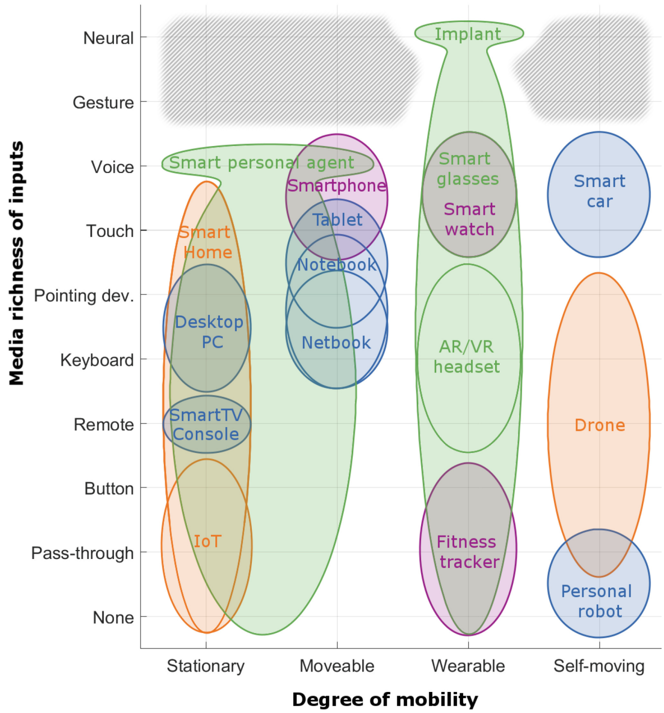

Fig. 3. Matrix of input and mobility dimensions (adapted from [1]).

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* evolved from 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 – thereby such a class represents an excellent high-level overview yet a poor low-level discriminating power. 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.

Differences in the device classes can also be explained with regard to *media richness theory* (MRT). MRT describes a corridor of effective communication with matching levels of message ambiguity and media richness [37]. 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 possibilities for direct user input but also give not much feedback in return. Notebooks allow for medium levels of input richness through keyboard

Table 1. App-enabled device classes and their position in the continuum.

Device class	Input richness	Output richness	Degree of mobility
AR/VR headset	Remote - Pointing	Augmented reality	Wearable
Console	Remote	Small - Huge	Stationary
Desktop PC	Keyboard - Pointing	Large - Huge	Stationary
Drone	Pass-through - Pointing	None - Pass-through	Self-moving
Fitness trackers	None - Button	Pass-through - Tiny	Wearable
Implant	Neural	*any*	Wearable
	any	Neural	Wearable
IoT	None - Button	None - Pass-through	Stationary
Netbook	Keyboard - Pointing	Medium	Moveable
Notebook	Keyboard - Touch	Large	Moveable
Personal robot	None - Pass-through	None - Voice	Self-moving
Smart car	Touch - Voice	Medium - Large	Self-moving
Smart glasses	Touch - Voice	Augmented reality	Wearable
Smart home	None - Touch	None - Tiny	Stationary
Smart personal agent	Voice	None - Voice	Stationary-Moveable
	None - Voice	Voice	Stationary-Moveable
Smartphone	Touch - Voice	Small	Moveable
Smart TV	Remote	Huge	Stationary
Smartwatch	Touch - Voice	Pass-through - Tiny	Wearable
Tablet	Pointing - Touch	Medium	Moveable

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 user interactions.

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, output capabilities develop in two directions. On the one hand, screen sizes tend to diminish, from small screens on *smartphones* to very limited *fitness tracker* screens and screen-less *drones*. On the other hand, output capabilities become richer and overcome traditional screen-based approaches due to recent technological developments enabling intangible outputs, for instance *augmented/virtual reality (AR/VR) headsets*. It can also be observed that device classes with a high degree of mobility are more variable and occupy larger spaces of the continuum. This is potentially caused by a fragmentation into various domains of application, or their novelty of appearance with insufficient time to establish wide-spread interaction patterns. 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, Fig. 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.

MRT also partly explains why there are areas in the continuum with no assigned device class. Rich forms of user input such as gestures overcomplicate interactions for devices that have just small screens and therefore are typically equipped with limited sensing and processing resources. On the other extreme, devices with barely a few buttons do not provide sufficiently flexible input capabilities to manipulate large screens (such as several fingers multi-touch on a small smart watch). Of course, empty spaces in the taxonomy might also be caused by a lack of technological progress or use cases so far. Thus, they might actually be filled by future devices, or existing classes might "stretch" into these areas. For example, voice interfaces just recently emerged as mainstream technology in various devices from smartphones to smart home applications but augmented reality devices are still an active field of research. In general, with the evolution and differentiation of input media, existing device classes might extend towards further areas or even converge. For example, consider convertibles, such as the Lenovo Yoga Book, which represent hybrid devices between keyboard-based *netbooks* and touch-optimized *tablets* utilizing docking or folding mechanisms. Also, the evolution of one device class might render another obsolete; this can currently be observed with smartwatches cannibalizing the market for fitness trackers with more advanced input and output capabilities.

4 Discussion

The field of modern mobile computing does not show signs of less rapid progress. It, thus, is likely that amendments will need to be made. Additionally, we will need to keep updating the taxonomy once it has been acknowledged by the scientific community. Moreover, a taxonomy should be appealing for the use by practitioners, particularly in a field where scientific research and technological progress go hand-in-hand. Therefore, this section presents ideas for discussion that go beyond the narrower focus of Sect. 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 resolution, raw computing power, touch screen availability) or usage (e.g., business, entertainment, sports, or communication) fail to provide clear criteria for a taxonomy. While they may even pose discriminatory power, they do not necessarily help with forming adequate classes of devices. In particular, a fast adaptation and convergence of available technologies could be observed in the past years. Using simple hardware features for categorization would thus be prone to quickly becoming obsolete. To give some examples: so-called *phablets* blur the lines between smartphones and tablets; cameras with resolutions a few years ago only imaginable in professional photography equipment now are routinely built into many mobile devices; and gyroscope sensors have found wide-spread adoption in a variety much mobile hardware for a variety of purposes.

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, touch-based screens are usual interfaces both for tablets and for the infotainment systems within 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. [4] 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 profoundly gain more discriminatory power. The added complexity would not be justified as the underlying assumption of increasing processing complexity is to some extent already encoded in the richness of inputs and outputs.

An alternative or possibly additional means for categorization is a *graph*, more specifically a *tree*. This way, categorization would get a hierarchical character that could for example honour development history and be subdivided. Additionally, this representation would be well suited to reveal similarities in particular features. If shown as a *polytree* such as sketched in Fig. 4, even complex dependencies could be displayed (a smart watch, for example as a combination of wrist-worn fitness trackers and basic smartphone functionality). However,

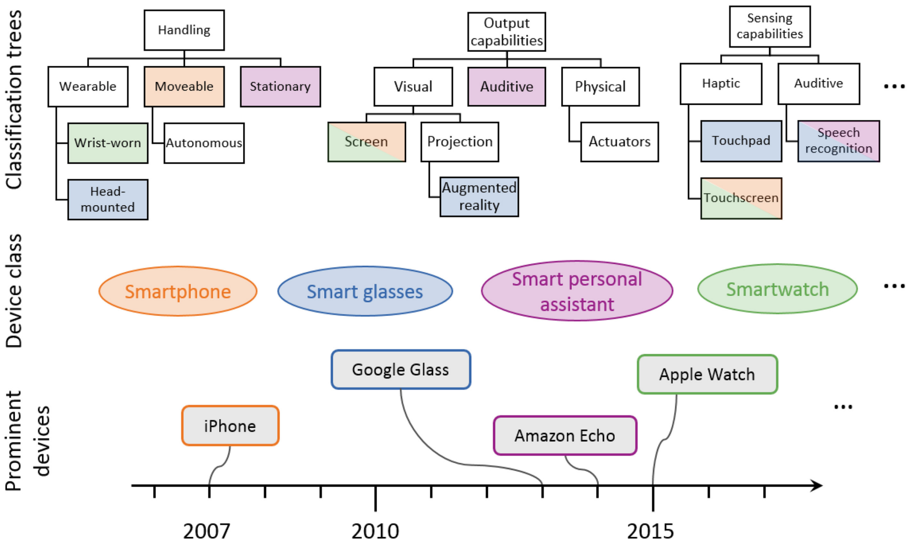

Fig. 4. Exemplary alternative classification approach.

while a tree representation surely is charming for its depiction of dependencies and historical connections, it poses far less discriminatory power than the taxonomy we designed. Blending both possibilities, for instance using a multi-layered representation of several trees that show the connection between device classes from different perspectives or for various criteria, would become too complex to be practical.

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. This is an ongoing task.

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. In any case, taxonomies that are used in any not entirely static field ought to evolve.

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 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. Moreover, commercial (and, similarly, military) devices might be derivatives of consumer hardware that has been "hardened" and more extensively tested. 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 regarding them is scarce, we deem the current limitation justified. Additionally, if kinds of devices are seldom addressed in writing, including them in a taxonomy arguably is superfluous anyway.

Fourthly, it should be scrutinized how the taxonomy can be provided in a form that is useful for researchers *and* for practitioners. Most scientists know taxonomies for research topics enforced by publication outlets.[5] 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 harder 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. We think that our taxonomy should allow to put each device into exactly one class – choosing several applicable classes might be practical for the above named paper-theme categories, but we do not deem it practical for the purpose of our taxonomy.

The four discussion points have also illustrated the limitations of our work. Besides these issues that need to be worked on, an eventual verification of the taxonomy is mandated. Our planned work to further on this topic is sketched in the next section.

It would also be possible to develop the taxonomy towards an ontology (cf. [39]), possibly resulting in automated categorization aid which would also take ideas from the alternative polytree-based categorization as discussed in the previous section. For an unknown device, a decision tree could be traversed, leading to a prediction which kind of device is at hand. However, this would require rich semantic data (to allow inference), and it is currently not clear whether such an ontology would be considered to have much more value than the taxonomy already possesses.

[5] An example is be the ACM Computing Classification System, firstly presented in 1964 and revised in 1991, 1998, and 2012 [38] (cf. also [39]).

5 Conclusion and Outlook

With this article, we have proposed a taxonomy for app-enabled devices. It builds on a position paper presented in April 2017 – and it remains the first such work. The taxonomy is based on three dimensions: the media richness of inputs and of outputs, and the degree of mobility. Examined separately, each dimension is relatively simple. In combination, they provide high discriminatory power. This becomes particularly evident when categorizing the current device landscape. We have provided figures that support this assessment throughout this paper. In general, it has proven to be much easier to use "flat" representations of two dimensions at a time than to render a 3D model – at least for publication.

The presented taxonomy can be considered as a milestone, and we deem it to be static for now. Undoubtedly, progress in the field will mandate future changes, but these will rather lead to a new *version* of the taxonomy than to a *new* taxonomy. Nonetheless, this article should still act as an invitation for discussion. After all, the taxonomy is but a step towards a more unified view of mobile computing and a solidified theoretic base in this field. Moreover, future work will need to continue with keeping a systematic overview of app-enabled devices.

A better theoretic understanding of mobile computing, producible advice for practice, and word towards unified development mark the pillars of our future work. As a part of this, we will use the taxonomy and also put it up for further discussion, e.g. as part of conference talks. Additionally, we will now reach out to our partners from practice and ask them for an assessment. If the taxonomy will become well adopted, empirical work should follow.

As we already wrote in the position paper, we do not hope for our work to become "yet another computer science taxonomy". Therefore, we hope that this article can illustrate the usefulness of a taxonomy and spark the interest for employing it.

References

1. Rieger, C., Majchrzak, T.A.: Conquering the mobile device jungle: towards a taxonomy for app-enabled devices. In: Proceedings of the 13th International Conference on Web Information Systems and Technologies (WEBIST), pp. 332–339. SciTePress (2017)
2. Statista Inc.: Global smartphone shipments forecast from 2010 to 2020 (2016). https://www.statista.com/statistics/263441/
3. Chuah, S.H.W., Rauschnabel, P.A., Krey, N., Nguyen, B., Ramayah, T., Lade, S.: Wearable technologies: the role of usefulness and visibility in smartwatch adoption. Comput. Hum. Behav. **65**, 276–284 (2016)
4. Coppola, R., Morisio, M.: Connected car: technologies, issues, future trends. ACM Comput. Surv. **49**(3), Article No. 46 (2016)
5. Apple Inc.: Apple reinvents the phone with iPhone (2007). http://www.apple.com/pr/library/2007/01/09Apple-Reinvents-the-Phone-with-iPhone.html

6. Apple Inc.: iPhone app store downloads top 10 million in first weekend (2008). http://www.apple.com/pr/library/2008/07/14iPhone-App-Store-Downloads-Top-10-Million-in-First-Weekend.html
7. Biørn-Hansen, A., Majchrzak, T.A., Grønli, T.M.: Progressive web apps: the possible web-native unifier for mobile development. In: Proceedings of the 13th International Conference on Web Information Systems and Technologies (WEBIST), pp. 344–351. SciTePress (2017)
8. Heitkötter, H., Hanschke, S., Majchrzak, T.A.: Evaluating cross-platform development approaches for mobile applications. In: Cordeiro, J., Krempels, K.-H. (eds.) WEBIST 2012. LNBIP, vol. 140, pp. 120–138. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36608-6_8
9. El-Kassas, W.S., Abdullah, B.A., Yousef, A.H., Wahba, A.M.: Taxonomy of cross-platform mobile applications development approaches. Ain Shams Eng. J. **8**, 163–190 (2015)
10. Weiser, M.: The computer for the 21st century. Sci. Am. **265**, 94–104 (1991)
11. Statista Inc.: Global smart tv unit sales from 2014 to 2018 (in millions) (2017). https://www.statista.com/statistics/540675/global-smart-tv-unit-sales/
12. Statista Inc.: Smartwatches (2017). https://www.statista.com/study/36038/smartwatches-statista-dossier/
13. Ibarra-Esquer, J.E., González-Navarro, F.F., Flores-Rios, B.L., Burtseva, L., Astorga-Vargas, M.A.: Tracking the evolution of the internet of things concept across different application domains. Sensors **17**, 1379 (2017)
14. Heitkötter, H., Majchrzak, T.A., Kuchen, H.: Cross-platform model-driven development of mobile applications with MD2. In: Shin, S.Y., Maldonado, J.C. (eds.) Proceedings of the 28th Annual ACM Symposium on Applied Computing (SAC), SAC 2013, pp. 526–533. ACM (2013)
15. Jones, C., Jia, X.: The AXIOM model framework: transforming requirements to native code for cross-platform mobile applications. In: Ferreira Pires, L. (ed.) 2nd International Conference on Model-Driven Engineering and Software Development. IEEE (2014)
16. Chauhan, J., Mahanti, A., Kaafar, M.A.: Towards the era of wearable computing? In: Proceedings of 2014 CoNEXT on Student Workshop, CoNEXT Student Workshop 2014, pp. 24–25. ACM (2014)
17. Busold, C., Heuser, S., Rios, J., Sadeghi, A.-R., Asokan, N.: Smart and secure cross-device apps for the internet of advanced things. In: Böhme, R., Okamoto, T. (eds.) FC 2015. LNCS, vol. 8975, pp. 272–290. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-47854-7_17
18. Dageförde, J.C., Reischmann, T., Majchrzak, T.A., Ernsting, J.: Generating app product lines in a model-driven cross-platform development approach. In: 49th Hawaii International Conference on System Sciences (HICSS), pp. 5803–5812 (2016)
19. Jesdabodi, C., Maalej, W.: Understanding usage states on mobile devices. In: Proceedings of 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing, UbiComp 2015, pp. 1221–1225. ACM (2015)
20. Chauhan, J., Seneviratne, S., Kaafar, M.A., Mahanti, A., Seneviratne, A.: Characterization of early smartwatch apps. In: 2016 IEEE International Conference on PerCom Workshops 2016 (2016)
21. Google Inc.: Android wear (2016). https://android.com/wear
22. Apple Inc.: WatchOS (2016). https://www.apple.com/watchos/
23. LG Electronics: WebOS for LG Smart TVs (2016). http://www.lg.com/uk/smarttv/webos

24. The Linux Foundation: Tizen (2016). https://www.tizen.org
25. Neate, T., Jones, M., Evans, M.: Cross-device media: a review of second screening and multi-device television. Pers. Ubiquit. Comput. **21**, 391–405 (2017)
26. Singh, K., Buford, J.: Developing WebRTC-based team apps with a cross-platform mobile framework. In: 13th IEEE Annual Consumer Communications and Networking Conference, CCNC 2016 (2016)
27. Queirós, R., Portela, F., Machado, J.: Magni - a framework for developing context-aware mobile applications. In: Rocha, Á., Correia, A.M., Adeli, H., Reis, L.P., Costanzo, S. (eds.) WorldCIST 2017. AISC, vol. 571, pp. 417–426. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-56541-5_43
28. Koren, I., Klamma, R.: The direwolf inside you: end user development for heterogeneous web of things appliances. In: Bozzon, A., Cudre-Maroux, P., Pautasso, C. (eds.) ICWE 2016. LNCS, vol. 9671, pp. 484–491. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-38791-8_35
29. Humayoun, S.R., Ehrhart, S., Ebert, A.: Developing mobile apps using cross-platform frameworks: a case study. In: Kurosu, M. (ed.) HCI 2013. LNCS, vol. 8004, pp. 371–380. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39232-0_41
30. Sommer, A., Krusche, S.: Evaluation of cross-platform frameworks for mobile applications. Lecture Notes in Informatics (LNI) P-215, pp. 363–376 (2013)
31. Dalmasso, I., Datta, S.K., Bonnet, C., Nikaein, N.: Survey, comparison and evaluation of cross platform mobile application development tools. In: 2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC) (2013)
32. Rieger, C., Majchrzak, T.A.: Weighted evaluation framework for cross-platform app development approaches. In: Wrycza, S. (ed.) SIGSAND/PLAIS 2016. LNBIP, vol. 264, pp. 18–39. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46642-2_2
33. Lahiani, H., Kherallah, M., Neji, M.: Vision based hand gesture recognition for mobile devices: a review. In: Abraham, A., Haqiq, A., Alimi, A.M., Mezzour, G., Rokbani, N., Muda, A.K. (eds.) HIS 2016. AISC, vol. 552, pp. 308–318. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-52941-7_31
34. Bhowmik, A.K.: 39.2: Invited paper: natural and intuitive user interfaces: technologies and applications. In: SID Symposium Digest of Technical Papers, vol. 44, pp. 544–546 (2013)
35. Hintze, D., Hintze, P., Findling, R.D., Mayrhofer, R.: A large-scale, long-term analysis of mobile device usage characteristics. In: Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, vol. 1, pp. 1–21 (2017)
36. Gubbi, J., Buyya, R., Marusic, S., Palaniswami, M.: Internet of Things (IoT): a vision, architectural elements, and future directions. Future Gener. Comput. Syst. **29**, 1645–1660 (2013)
37. Daft, R.L., Lengel, R.H., Trevino, L.K.: Message equivocality, media selection, and manager performance: implications for information systems. MIS Q. **11**, 355 (1987)
38. ACM Computing Classification System ToC (2015). http://www.acm.org/about/class
39. Cassel, L.N., Palivela, S., Marepalli, S., Padyala, A., Deep, R., Terala, S.: The new ACM CCS and a computing ontology. In: Proceedings of 13th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL), pp. 427–428. ACM (2013)