The potential of smartwatches to support mobile industrial maintenance tasks

Jens Ziegler, Sebastian Heinze, Leon Urbas
Technische Universität Dresden
Chair of Process Control Systems Engineering
{jens.ziegler, sebastian.heinze1, leon.urbas} @tu-dresden.de

Abstract

This paper presents a study on the potential of today's smartwatches as a complementary user interface to mobile support systems for industrial maintenance tasks. Starting from challenges, usage scenarios and use cases for the valuable use of smartwatches in an industrial context, the various possibilities of information delivery, task support and process control using smartwatches are evaluated and checked against basic ergonomic requirements on data pre-processing, user interface design and the hardware equipment. A prototypical implementation of a support system for industrial maintenance tasks illustrates the applicability of the derived interaction concepts and design guidelines for smartwatch-equipped mobile support systems.

1. Introduction

2014 was the year of the wearables [1]. Especially smartwatches have become firmly established in the consumer market. Smartwatches are wearables with an appearance similar to wristwatches, but provide powerful computational resources, various sensors and wireless interfaces. They put significantly less workload when using compared to smartphones. The placement at the forearm is widely accepted by users. A smartwatch is usually within sight and within the reach of the opposite hand of the user, thus visibility and immediate accessibility are quite good. In addition, the device may be operated with an autonomous power supply in the near future [2]. It is therefore reasonable to thoroughly investigate the possible applications of smartwatches as an additional component in mobile support systems. Those systems are an integral part of a variety of mobile workflows (e.g. in warehouse logistics or freight shipping) and can lead, if used wisely, to a significant reduction of workload and an increase in the quality and availability of the work results [3].

Mobile support systems consist of at least one mobile device running a mobile application, and a remote, oftentimes distributed data backend. Both are at least intermittently wirelessly connected. The mobile devices are usually designed for operation in harsh environments. Such *rugged devices* are robust, tough and durable, but are still heavy and unwieldy, which puts a significant workload on the user.

Using a smartwatch as an additional user interface that continuously provides important pieces of information might improve the current situation. Smartwatches allow the system to provide important information and current messages permanently as required. It can also accept limited inputs to the mobile system with little effort. For viewing complex datasets or performing extensive interaction tasks, users may switch to the mobile application on the connected mobile device. Thus, the smartwatch needs to be highly integrated in the mobile support system in order to create a consistent and positive user experience.

However, it cannot be denied that the possible applications of smartwatches are limited by display size and the limited number of input means. Further, not every possible usage scenario is compatible with the paradigm of low-interaction, short-term and proactive use, which is the foundation of wearable interaction [4]. The work presented in this paper applies the use case of the IT-supported mobile maintenance of large chemical process plants to examine this holistic design challenge and to investigate the potential of smartwatches as a complementary user interface to mobile support systems for industrial use.

First, the underlying operating system (OS) and user interface design guidelines will be briefly introduced. Then, use cases and usage scenarios will be identified, and recommendations for the usable design of appropriate user interfaces will be given. The paper closes with the presentation of a prototypical implementation of a smartwatch-equipped mobile support system for industrial maintenance tasks that confirms the applicability of the presented design guidelines.

978-1-4673-7929-8/15/$31.00 ©2015 IEEE

2. IT-supported mobile maintenance

2.1. Context of use

The industrial maintenance of process plants is a demanding and versatile context of use. Maintenance staff working on-site in the plant maintains, repairs and overhauls the entire existing equipment and is thus responsible for the safe, correct and fault-free operation of the plant. The staff members use a variety of tools and usually wear personal protective equipment, which varies from simple helmet and protective gloves to chemical protective coverall, depending on the requirements of the actual task. They often work alone, sometimes even without contact to other people.

The environmental conditions may vary and fluctuate significantly. Typical factors are noise, vibration, changing and often adverse lighting conditions, dirt and a variety of coloring, lubricating or harmful substances, as well as a more or less rough terrain [5].

During their work the maintenance staff handles a variety of data, such as operating data for maintenance, performance indicators and messages from the system, as well as condition data and equipment histories. The staff members also generate data, such as reports on inspection results and taken maintenance measures. Some tasks also require data from other aspects than the maintenance, such as financial indicators and data, engineering data or process values of the plant. All these data can be obtained by plant asset management systems, and provided either as hard copy or using appropriate mobile IT [6].

2.2. Challenges for mobile information technology

Workflows in the area of mobile industrial maintenance are characterized by constant alternation between manual work on the system and interaction tasks with the mobile or on-site information technology (IT). Manual work is often strenuous and has usually higher priority than interaction tasks [7]. This means that minimizing the workload put on the user by using a mobile system must be a main design goal for mobile IT. The versatile workflows and the high autonomy of task scheduling makes it possible that the maintenance staff comprehensively optimizes their activities on-site with respect to overall processing time and workload distribution. Thus, a mobile system must be highly context-sensitive and adaptive to such kinds of optimization. Otherwise, the system would patronize the user, which is not acceptable [8]. Furthermore, the maintenance tasks are often dangerous and associated with high economic risks. This results in high demands on the robustness, durability and safety of the mobile systems, the quality of information as well as the usability of the system. Thus, high trustworthiness, reliability and usability are also core design objectives.

3. Background

3.1. The Google Android Wear operating system

Android Wear is a modified *Android* OS especially tailored for extra small screen sizes. It has been developed by the *Google Inc*. and is used today by several smartwatch vendors. It extends the capabilities of the mobile device by providing messages and simple *Action Controls* on the Bluetooth-connected smartwatch. *Android Wear* integrates the *Google Now* technology, which provides location-based services and a powerful speech recognition engine. *Android Wear* applications are developed with the same means as standard *Google Android* applications.

While *Google Android* currently dominates the market for smartphones with a market share constantly above 80 % (82.3 % as of 2014) [9], there is no definite trend towards one of the concurring platforms on the smartwatch market so far. However, *Android Wear* is the only OS that is commercially used by multiple vendors, which together cover almost 50 % of the smartwatch market [10]. These facts lead to the decision to use this operating system as the basis for the work presented, although the concepts presented are in principle independent of the underlying platform.

3.2. User interface design guidelines

The *Android Wear* user interface concept is based on the principles *simplicity of presentation*, *minimal interaction* and *high context sensitivity* [11]. Instead of explicit interaction, the smartwatch shall identify user needs for information or actions proactively based on the evaluation of the available context information. Then, the corresponding user interface shall be provided automatically and appropriate to the situation without interrupting the user in his work.

Google provides detailed design guidelines for *Android Wear* applications [11]. Besides the dial (*Watch Face*) and the *Home Screen*, the *Context Stream* is the main *Android Wear* user interface (UI) element. It provides all messages from the connected mobile device in a vertical list of *Cards*. *Standard Cards* show pieces of information in a uniform and consistent layout. *Single Action Controls* further provide a single *Action Button* with clear reference to the information displayed. A *Stack of Cards* groups several *Cards* with related content. The *Cue Card* is displayed whenever a voice command is expected. It also shows a list of all possible commands in full-screen mode as an alternative input means. It is also possible to create custom layouts, which are displayed in full-screen mode as well. *Cards* in the *Context Stream* are organized by the system, subsequent *Cards* or custom layouts can be arranged as two-dimensional matrices. These subsequent structures may provide additional information and control options for the related *Card*.

The application is operated by voice and direct interaction. Voice commands trigger *Android Intents*, and speech-to-text recognition is used for any text input. Swiping along the main axes allows users to navigate the *Cards*, to remove *Cards* from the *Context Stream*, and to scroll lists and displayed text. Navigation should be vertically first, and then horizontally.

Action Controls and list elements can be selected by tap gestures. A keyboard is not available by design. However, additional input techniques can be added, for example gesture-based input on the touch screen [12] or by moving the smartwatch [13]. The auto-ID functionality (e.g. NFC) provided by some devices enables interactions based on the principles of *Physical Mobile Interaction* [14]. Eventually, UI designers might want to bypass the *Android Wear* UI and develop applications using the standard *Google Android* UI elements.

4. Smartwatch-equipped mobile support systems for industrial maintenance tasks

4.1. State of the art

Early examples such as the *RUPUTER* [15] or the *Microsoft SPOT* [16] are evidence of nearly two decades of development and evolution. However, it was the availability of the first open-source, multi-vendor OS for wearables that finally constituted the breakthrough for smartwatches on the consumer market, most notably *Google Android Wear* and *Tizen*, which has been developed by the *Linux Foundation* [17]. In 2015, the operating systems *Apple Watch* and *WebOS for Smartwatches* followed.

In particular, in the field of healthcare, there are a number of applications available today, for example for the permanent monitoring of vital parameters or physical stress, or for controlling correct self-medication [18]. In the automotive industries, the integration of smartwatches into the automotive user interfaces and interactive vehicular applications makes good progress [19, 20]. There are also first projects in the area of enterprise software. For example, the SAP SE cooperates with the Samsung Corp. to extend the *SAP HANA cloud platform* with an additional user interface on a smartwatch. This project also relies on *Android Wear* [21]. The *Fraunhofer Institute* uses smartwatches as OPC UA clients to retrieve process data and provide simple machine control actions [22].

There is also a large body of scientific literature on innovative interaction means and techniques, such as the recognition of body gestures or facial expressions, fine-grained tactile interaction using the wrist contact. However, those techniques are not yet readily available, nor will they be in the near future. Therefore, the discussion of these future technologies is necessarily beyond the focus of this paper.

4.2. Usage scenarios and use cases

Smartwatches provide unique features that clearly set them apart from other mobile devices such as smartphones or tablets. A smartwatch is constantly available and easily observable, unmonopolizing and unobtrusive, easy to use and to carry along. The direct connection with the skin at the wrist allows for almost hands-free operation and immediate tactile feedback.

Based on these features, usage scenarios and use cases for mobile maintenance can be derived, where smartwatches show clear advantages compared to other devices mobiles. It is important to note that the Smartwatch does not need to replace other devices completely, but rather complement them in order to improve the overall user experience. In the following five basic use cases and typical usage scenarios are presented.

Use Case 1 – Explicit Information Push/Pull: The simplest use case for a smartwatch is to explicitly push or pull requested or created data. Pulled data are processed and displayed according to their structure and can be navigated and selected by the user. Pushed data are validated and sent to the data backend. Typical usage scenarios for pulling data are accessing files in folder structures (e.g. documentation folders), browsing parameter lists or organizing task execution. Typical usage scenarios for pulling data are modifying documents, changing parameters or reporting working results.

Use Case 2 – Implicit Information Push/Pull: More valuable than providing data on request is to inform the user proactively. A smartwatch can evaluate the current context of use, identify informational needs and automatically select the data to be displayed as well as the point in time when it should be displayed. It can provide easily graspable data with low complexity and actual relevance, but also data with higher complexity which is provided proactively in small portions. A smartwatch can also monitor the user and the current context, process and aggregate these data and automatically push gathered information to the data backend. Typical usage scenarios are receiving system notifications such as appointments, notifying the user about incoming work orders or notifying the data backend about an undesired health condition of the user.

Use Case 3 – Guidance: The smartwatch can also be used to guide the user to a location or through a procedure by regularly providing helpful information about the next necessary action. The selection and provision of relevant information can be automated based on contextual information (e.g. geolocation) or based on explicit user input. Typical usage scenarios are navigating to plant equipment or other staff or the step-by-step support of repair or overhaul activities.

Use Case 4 – Communication: The probably most common use case today is instant messaging. Especially if the user is mainly receiver of messages (e.g. in a group chat) reading incoming messages with a simple glance at the wrist requires much less effort than using other devices. Usage scenarios might be conversations between the on-site personnel and the plant operators, or between the personnel and actively communicating technical components (e.g. cyber-physical systems). Further, conversations can be logged and annotated with metadata such as time stamps, location data of context information, which can improve the safety management of the plant.

Use Case 5 – Monitoring: Monitoring and surveillance tasks over a longer period can also be supported very well by a smartwatch, if they require regular, but brief attention by the user. For example, the development of a value can be displayed permanently and thus checked regularly on the smartwatch. For this purpose, the value can also be aggregated and mapped to different events. The user then receives a notification as soon as an event occurs, and the screen might turn on automatically. One possible usage scenario is the use of the process monitoring on site, e.g. for the verification of the success of a cleaning action. Alternatively, the smartwatch can also monitor the users themselves using the available biosensors. Thus, the user or any other person can be immediately notified about abnormal vital parameters of the user (*health monitoring*), or if the user enters a security-related area (e.g. hazardous areas). Further, the effectiveness and efficiency of the user can be monitored as well. This can be of great benefit in the context of occupational health and safety management of the plant.

Use Case 6 – Control: Finally, smartwatches can also be used for very simple control actions. However, the complexity of the control action should be reduced to a minimum, since input on a smartwatch is comparatively error-prone and corrective means are very limited. Typical usage scenarios are the starting and stopping of actuators, the opening and closing of gates, or the logon and logoff of the user in certain areas or systems (e.g. using NFC technology). Further, operations such as clearance or emergency shutdown of equipment could also be automated.

4.3. Data formats and visualization patterns

The design of the UI layout and UI elements for smartwatches is based on the data to be displayed and their data structures. The basic data structures can be categorized based on their dimensionality [23]. Text documents, planar lists and forms may be regarded as one-dimensional data. These data can be reduced and split into smaller portions until each portion fits into a single *Card*. The *Cards* may then be navigated on the smartwatch using vertical swipe gestures.

Table, network and tree structures are two-dimensional data with solely categorical positional relationships of the data elements. These can be split into portions of data with two-dimensional relations, which can be displayed and navigated as a matrix using vertical and horizontal and vertical swipe gestures.

Maps and images are two-dimensional data with metric positional relationships. Thus, structural changes are only possible to a limited extent. These data are usually navigated by panning and zooming, which requires appropriate control options.

Models of real-world objects (e.g. CAD models) are three-dimensional data. For the presentation of these data, a smartwatch appears generally unsuitable. Complex abstract data structures can also have a multi-dimensional structure, which might be truncated by filtering or faceting, if possible.

Motion pictures, such as videos and animations have two spatial and an additional temporal dimension. It does not make sense, as a rule, to navigate along the spatial dimensions. If necessary, appropriate control options must be provided for navigating along the time axis, which should be reduced on smartwatches as much as possible. Audio data only have a temporal dimension. However, graphical control options might have to be provided as well.

4.4. Interaction concepts and design guidelines

Small amounts of data (e.g. reports, simple instructions or numeric representations of single process values) can be realized by *Standard Cards*, subsequent actions can be implemented by *Action Controls*. The *Cards* can either be organized automatically using the *Context Stream*, or by the developer using a *2D Picker*. Complex data sets or extensive interactions are not covered by the *Android Wear* design concept. If necessary, a custom layout needs to be realized to provide appropriate functionality. However, it is always preferable to truncate the data or the number of interactions, and to use standard UI elements.

Android Wear also provides means for displaying conversations (e.g. chats) using *Standard Cards* and control actions. To enter text, *Android Wear* only offers speech recognition. If other input techniques (e.g. a virtual keyboard) are considered necessary, they must be implemented.

Two-dimensional structures without metric positional relationships can be displayed *Standard Cards* and control actions and organized in a *2D Picker*. Two-dimensional structures with metric positional relationships should be displayed in full-screen mode with continuous panning and gradually zoom option. The illustration on very small screens must be generalized sufficiently in order to be recognizable by the user.

The display of motion pictures fundamentally contradicts the wearable interaction paradigm of short-term interaction and shall be avoided. Same holds for complex three-dimensional graphical representations which require complex input sequences to navigate the data.

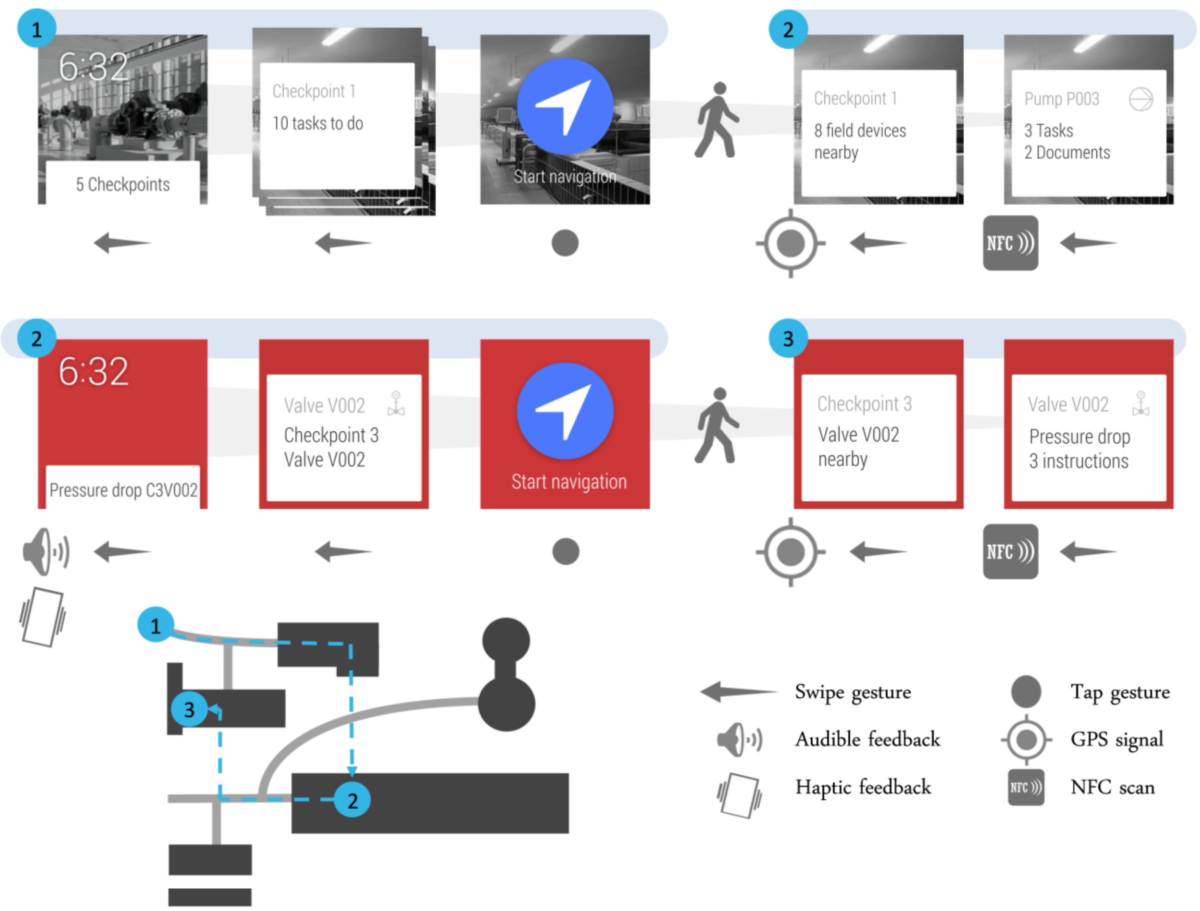

Figure 1. User interface design of the prototypical mobile support system. The maintenance tour is shown inset.

A special feature is the provision of geo-information and navigation means, which can be shown in a textual, iconographic or graphic manner. The navigation target can be entered by speech or selected from a list or a *2D Picker*. The calculated route can be displayed as a list containing all relevant landmarks, as a stream of navigation icons (e.g. arrows or traffic signs) shown in a single *Standard Card* or using a full screen map. If additional control actions are required in the map, a custom layout needs to be implemented.

Basically, the interaction with the smartwatch should be kept short-lasting and interruption-free with respect to the activity of the user. A single interaction should never take more than five seconds [11]. Swipe gestures should always use the entire screen, targets for tap gestures should always be presented on the largest possible area. The number of inputs should be reduced to a minimum.

In industrial applications, particularly the speech input is discussed controversially. Although a satisfactory detection performance is achievable today, the remaining error rate, the rudimentary correction means and the fact that speech today is analyzed remotely in order to provide sufficient computational power, limit the application of this input technique in industrial applications.

For this reason, additional input means should be provided if possible, e.g. selection lists or other input elements that can be operated with swipe gestures. If these input techniques are not sufficient, the activity should be continued on the connected mobile device. Prototypical implementation

The recommendations presented in the previous section have been used to implement a smartwatch-equipped mobile support system for industrial maintenance tasks. This system consists of a smartphone running the mobile application, a smartwatch and a *Linked Data Server* as data backend. This Server provides all available engineering data, process data, management data and maintenance data of the digital plant.

As an example, Figure 1 shows two different scenarios in a mobile maintenance tour (the tour map is shown inset on the lower right corner). The users start their maintenance tour (point 1 on the map) and thus get a short overview of the checkpoints of the day. With a swipe gesture to the left they navigate to the *2D Picker* that provides further information on each checkpoint. With another swipe gesture they navigate to an *Action Control* that allows the users to start the navigation to the selected checkpoint. They follow the instructions of the smartwatch until they arrive at checkpoint 1 (point 2 on the map). Once the system recognizes that the users have arrived (e.g. using the GPS data), they will get a notification about eight field devices that need to be maintained. Instead of explicitly select one of them in the subsequent *2D Picker*, the user scans the NFC tag of pump P003 with their smartwatch. Immediately, the smartwatch displays the tasks and documents associated with this particular device. Now, they can start with the maintenance.

After a while, the users get a notification from the smartwatch. A pressure drop was recognized in valve V002 at the checkpoint 3. The red background color and the audible and tactile feedback of the smartwatch indicate a critical fault with processing priority. Thus, the users interrupt their current activity and switches to the new task. With a swipe gesture the user navigates to a *Standard Card* that provides additional information about the incident. With another swipe gesture they again navigates to the navigation function, which guides them to checkpoint 3 (point 3 on the map).

Once the users get close to valve V002, they get a notification. Again, they identify the valve using the NFC-reader of the smartwatch, this time to get the three possible therapeutic measures. The users can select from them in a *2D Picker* to get detailed instructions on how to eliminate the fault. The users can now perform their troubleshooting until the problem is solved. By confirming the successful troubleshooting the corresponding *Card* is closed and the application automatically returns to the previous *Card*.

These two comparatively simple scenarios perfectly demonstrate the potential of smartwatches to support maintenance staff on-site solely based on the standard *Android Wear* UI elements. They also show how these structurally quite similar dialogues provide the user completely different information and thus support the different tasks in an optimal manner based on contextual information about the location of the user and the target, and about the priority of the task.

5. Summary

In this article, the potential of smartwatches to support mobile industrial maintenance tasks has been examined. Therefore, the underlying operating system and user interface design guidelines have been introduced briefly, and use cases and usage scenarios have been identified. Afterwards, recommendations for the usable design of appropriate user interfaces have been developed and elucidated using a prototypical implementation of a smartwatch-equipped mobile support system for industrial maintenance tasks.

It has been shown that smartwatches can significantly reduce the workload put on the user for a number of usage scenarios, and that appropriate user interfaces can be implemented to a large extent without any major changes or additions to the *Android Wear* UI elements.

References

[1] Metz, R. (2014). *2014 in Mobile: Wearable Gadgets Galore*. URL: www.technologyreview.com/news/533691/2014-in-mobile-wearablegadgets-galore/ (last visited 04.04.2015)

[2] Jung, W. S., Lee, M. J., Kang, M. G., Moon, H. G., Yoon, S. J., Baek, S. H., & Kang, C. Y. (2015). Powerful curved piezoelectric generator for wearable applications. In *Nano Energy*, 13, 174-181.

[3] Ziegler, J., Urbas, L., Lenz, H., & Richter, H. (2014). App-based system diagnosis using mobile information systems. In *Emerging Technology and Factory Automation* (ETFA 2014), 1-7. IEEE.

[4] Dvorak, J. L. (2007). *Moving Wearables into the Mainstream: Taming the Borg*. Springer.

[5] Wittenberg, C. (2008). Is multimedia always the solution for human-machine interfaces? – A case study in the service & maintenance domain. In *Systems, Signals and Image Processing* (IWSSIP 2008), 393-396. IEEE.

[6] VDI/VDE 2651 – *Plant Asset Management (PAM) in the process industry - Definition, model, task, benefit*. Beuth Verlag, 2008.

[7] Boronowsky, M., Nicolai, T., Schlieder, C., & Schmidt, A. (2001). Winspect: A case study for wearable computing-supported inspection tasks. In *Wearable Computers 2001*, 163-164. IEEE.

[8] Spiekermann, S., & Pallas, F. (2006). Technology paternalism–wider implications of ubiquitous computing. In *Poiesis & praxis*, 4(1), 6-18.

[9] statista (2015). Forecast on the market shares of smartphone operating systems sales worldwide in 2014 and 2018.

[10] statista (2015). Market shares of the leading manufacturers in the turnover with Smart Watches / Fitness bracelets worldwide in 2014.

[11] Google Developers Page. URL: developer.android.com/design/wear/creative-vision (last visited 04.04.2015)

[12] Microsoft Research: Analog Keyboard for Android Wear. URL: research. microsoft.com/en-us/um/redmond/projects/analogkeyboard/ (last visited 04.04.2015)

[13] Xiao, R., Laput, G., & Harrison, C. (2014). Expanding the input expressivity of smartwatches with mechanical pan, twist, tilt and click. In *Human factors in computing systems*, 193-196. ACM.

[14] Rukzio, E., Leichtenstern, K., Callaghan, V., Holleis, P., Schmidt, A., & Chin, J. (2006). An experimental comparison of physical mobile interaction techniques: Touching, pointing and scanning. In *Ubiquitous Computing* (UbiComp 2006). 87-104. Springer.

[15] OnHand PC. URL: www.pconhand.com (last visited 04.04.2015)

[16] Bonnert, E.; Kuri, J. (2002). Comdex: Bill Gates dreams of the magic of software URL: www.heise.de/newsticker/meldung/Comdex-Bill-Gates-traeumt-vom-Zauber-der-Software-70577.html (last visited 04.04.2015)

[17] Nickinson, P. (2014). The best Android Wear smartwatch. URL: www.android central.com/best-android-wear-smartwatch (last visited 04. 04. 2015)

[18] Swan, M. (2012). Sensor mania! The internet of things, wearable computing, objective metrics, and the quantified self 2.0. In *Journal of Sensor and Actuator Networks*, 1(3), 217-253.

[19] Ingraham, N. (2015). A new LG watch was just teased at Audi's CES press conference. URL: www.theverge.com/2015/1/6/7504199/a-newlg-smartwatch-was-just-teased-at-audis-ces-press-conference (last visited 04.04.2015)

[20] Johnson, M. (2015). Hyundai Blue Link to debut Smartwatch App with voice recognition at 2015 Consumer Electronics Show and Pepcom's Digital Experience. Hyundai press release.

[21] Hernandez, P. (2014). Samsung and SAP Team for Enterprise Wearable Apps. URL: www.datamation.com/mobile-wireless/samsung-and-sap-team-for-enterprise-wearable-apps.html (last visited 04.04.2015)

[22] Fraunhofer IOSB: SmartWatch now available for production. URL: www.iosb.fraunhofer.de/servlet/is/50420/ (last visited 04.04.2015)

[23] Shneiderman, B. (1996). The eyes have it: A task by data type taxonomy for information visualizations. In *Visual Languages 1996*, 336-343. IEEE.