Sharing Vital Signs between Mobile Phone Applications

Walter Karlen[1,2], *Member, IEEE*, Guy A Dumont[1], *Fellow, IEEE*, and Cornie Scheffer[2], *Member, IEEE*

Abstract— We propose a communication library, ShareVital-Signs, for the standardized exchange of vital sign information between health applications running on mobile platforms.

The library allows an application to request one or multiple vital signs from independent measurement applications on the Android OS. Compatible measurement applications are automatically detected and can be launched from within the requesting application, simplifying the work flow for the user and reducing typing errors. Data is shared between applications using intents, a passive data structure available on Android OS. The library is accompanied by a test application which serves as a demonstrator.

The secure exchange of vital sign information using a standardized library like ShareVitalSigns will facilitate the integration of measurement applications into diagnostic and other high level health monitoring applications and reduce errors due to manual entry of information.

I. INTRODUCTION

The popularity of mobile devices and the easy access to public software repositories is the ignition of a revolution in health care (mHealth). Home and self-monitoring using smart phones is already prominent with athletes and wellness conscious smart phone users and it is expected that mobile devices will play an important role in home care of chronic disease patients and in the diagnosis and treatment of infectious diseases in under-resourced areas [1]. Various health parameters are already recorded or displayed with smart phones using custom applications. The phones' connectivity options like USB, WiFi and Bluetooth allow for a connection to external medical sensors, such as glucose meters [2], blood pressure monitors [3] or pulse oximeters [4]. Integrated sensors can be used to extract vital signs, for example heart rate and oxygen saturation with the phone's camera [5].

Many challenges for the widespread adoption of mHealth remain. For example, there is a lack of standardization through regulatory bodies. Another risk to consider with mHealth is data security, privacy, and confidentiality and further research on security options is needed [6]. These challenges could be partly addressed by implementing open architectures and a standardization of protocols [7], supporting interoperability between devices, platforms and health applications.

We propose a standardized communication library, Share-VitalSigns, for the exchange of vital sign information be-

tween health applications running on the Android OS platforms. The library allows an application to request one or multiple vital signs. As a reply to this request, the OS will supply a list of external applications installed on the same device that support the ShareVitalSigns protocol and can provide the requested vital sign. The user can launch the measurement application of his choice and start a measurement. Upon completion of the measurement with the external application, the obtained vital sign is returned and can be further processed by the primary application.

The exchange of vital sign information using a standardized library like ShareVitalSigns will facilitate the integration of measurement applications into diagnostic and other health monitoring applications, simplifying the work flow required for patient vital sign measurement. Also, the standardization could increase adoption and consequently enlarge access to third party health measurement applications and devices. In return, this will allow developers of diagnostic applications to focus on algorithms and content.

II. SPECIFICATIONS

The specifications for the ShareVitalSigns library were defined to allow a modular communication of vital signs between Android applications. The requirements were that the library could be deployed into any existing or new Android (version 2.4 and higher) application without great effort by the software developer. Consequently, the Android software development kit and its native Java programming language were used for the library development. The library was to enable health applications to either request (as host application) or provide (as measurement application) one or multiple vital signs without the need of leaving the host application or manually copying or editing values. In addition, the host application had to be aware which applications installed on the device (if any) are capable of providing the desired vital signs. To facilitate quality management, the vital sign sent by the measurement application was to be accompanied by a confidence measure describing the quality of the performed measurement. Further, the communication of information between applications should respect the privacy of users.

III. RESULTS

The developed ShareVitalSigns library provides functions to request, send and receive vital signs from two independent applications installed on the same Android device. Two sets of functions are available, one to be used by the host application that sends the vital sign request, the other for the application that measures the vital signs. The library is accompanied by two test applications that either request
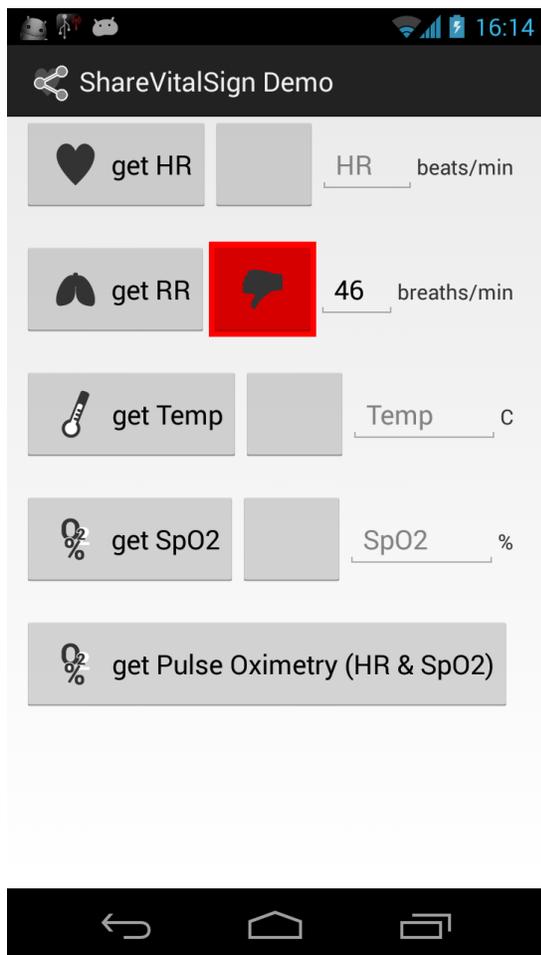
Fig. 1. ShareVitalSigns demo receiver application. Buttons allow for the request of vital signs such as heart rate (HR), respiratory rate (RR), temperature (Temp), blood oxygen saturation (SpO$_2$) or their combination. The returned value from the measurement application is displayed in a text field and the confidence is translated into a color code with a "thumbs up/down" sign.

or provide vital signs. They serve as a demonstration for developers on how to use the library functions and allow testing of ShareVitalSigns integration in new applications on target hardware devices (Figure 1). The library and demo applications are open source and available through the library's website[1].

Currently, the library supports five vital signs and any combination of these: heart rate, respiratory rate, oxygen saturation, temperature, systolic blood pressure and diastolic blood pressure. Vital signs were encoded as integers with values of the power of 2 in order to allow combinations (e.g. pulse oximeter can provide heart rate and oxygen saturation, blood pressure can provide systolic and diastolic blood pressure) using the bitwise **Or** operator (Table I).

The call of external applications is performed by creating an intent, an inherent capability of the Android OS that creates a passive data structure that contains a description of an action to be performed and allows the passing of

[1]ShareVitalSigns library website: http://sharevitalsigns.ecemgroup.org

information in both directions [8]. Intents can launch new activities. For example, intents are used to record a picture by opening the native camera application or to send a message using the email application. For handling intents containing vital signs, a custom dictionary defining the vital sign intent filters was created. The ShareVitalSigns intent is implicitly invoked by the host application. This allows the OS to search for applications on the device that can provide the requested vital signs (Figure 2). In order to advertise to other applications its vital sign measurement capability, a measurement application has to declare the type of vital signs with intent filters placed in the applications manifest file. The XML structure for ShareVitalSigns intent filters require, like any other intent filter on Android, a <action>, <data> and <category> field. The implicit nature of the intent requires the category to be set to DEFAULT. The <action> tag defines the type of vital sign that is provided. It is defined as the library package name "org.ecemgroup.sharevitalsigns.", followed by "MEASURE" and the vital signs key (Table I). Multiple <actions> can be defined in the same intent filter, for example pulse oximetry:

```
<intent-filter>
  <action android:name=
"org.ecemgroup.sharevitalsigns.MEASURE.HR"
/>
  <action android:name=
"org.ecemgroup.sharevitalsigns.MEASURE.SPO2"
/>
  <action android:name=
"org.ecemgroup.sharevitalsigns.MEASURE.PO"
/>
  <category android:name=
"android.intent.category.DEFAULT" />
</intent-filter>
```

The host application declares which intents the application is requesting by adding

```
<uses-permission android:name=
"org.ecemgroup.sharevitalsigns.MEASURE.
$VITALSIGNKEY$"/>
```

to the manifest file.

The vital signs request and answer structures are encoded as extra fields of the intent. The extra fields for requests consist of a key-value pair with the key "measure" and the value encoding the vital sign to be measured. Two new key-value pairs are created by the measurement application for each vital sign requested. The first pair contains the vital sign (float) and the second pair contains a matching confidence measure for the corresponding vital sign. The confidence value is an integer that can range from 0 to 100 and describes the reliability of the measured vital sign, where 100 is the highest confidence possible.

We have implemented and tested the library on two applications that provide vital signs, 1) *RRate*, an application that measures respiratory rate by registering taps on the screen [9]. and 2) the *PhoneOx*, an application that records heart rate and oxygen saturation from an external, commercial pulse oximeter [4]. The requesting and receiving functions have been embedded in the *ClinicalGuide* [10], an application to
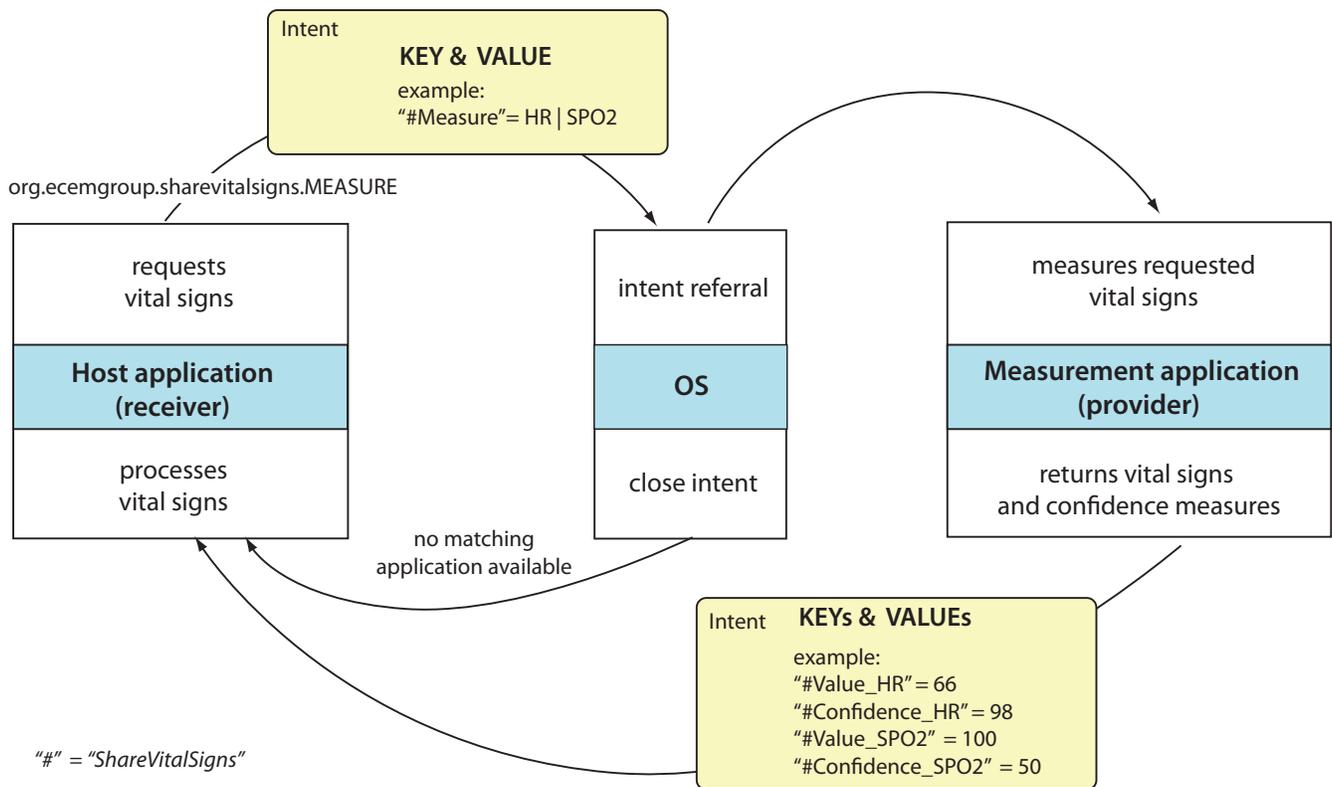
Fig. 2. Steps to request a vital sign using the ShareVitalSigns library. The host application is opening an intent with a measure field containing the requested vital sign. The Android OS then provides a list of applications that provide the requested vital sign. If an application is chosen, the intent is referred to the chosen measurement application. After the measurement, the intent is returned with new key and value pairs for the measured vital signs.

guide community health workers through the assessment of patients using the Integrated Management of Childhood Illness guidelines [11]. These guidelines provide algorithms for the assessment of children with cough or difficult breathing and suspected pneumonia. A key parameter for the diagnosis of pneumonia is the accurate measurement of respiratory rate. The questionnaire in *ClinicalGuide* has a field for manual respiratory rate entry, but also provides a button that requests respiratory rate from an external application.

IV. DISCUSSION

The developed ShareVitalSigns library allows applications to either request and/or share vital sign data on the Android platform. The library makes use of the established intent functions. The library is limited to 32 simultaneous vital signs that can be requested, which is sufficient for most diagnostic applications. Besides the vital sign, the transmitted packets also provide fields for confidence information on the respective vital sign. Submitting other extra fields (e.g. measurement settings) is possible, allowing for expansion of the library in future versions. The ShareVitalSigns library will allow developers of biomedical measurement applications to build small measurement applications that can easily communicate information with higher level applications, such as diagnostic or fitness applications, and share vital signs independent of internet connectivity. Detaching measurement applications from high level applications will allow for the diversification of the choice of measurement applications and provide access to vital sign measures in other scenarios, not in the scope of the original measurement application. This has the potential to broaden the landscape of medical monitoring applications.

Manual data entry of medical information can be prone to errors [12]. Acquiring vital signs from within the diagnostic application can reduce such errors. Other than the elimination of typographical errors and incorrect data entry, using the ShareVitalSigns library will allow the logging and tracing of how the vital signs were obtained. Also, the direct transmission of vital signs and other measurement data between applications can reduce the time to complete the task.

We have included a field for quality measurement for each vital sign measured and transmitted. While this can improve quality and avoid misinterpretation of vital signs, it cannot guarantee accuracy. As with data obtained from external devices, e.g. via a Bluetooth connection, the host application has no control on how the vital sign is calculated by the measurement application. The implicit intent call leverages this problem by allowing the user to select the measurement application of choice. However, this requires the user to have knowledge of the measurement application procedures. In addition, it remains the users responsibility to install ShareVitalSigns supported applications from only trustworthy developers or manufacturers. Future work could include a certification system that allows measurement appli-

TABLE I

ENCODING OF VITAL SIGNS FOR DIFFERENT LEVELS OF THE SHAREVITALSIGNS LIBRARY. ACTION NAMES ARE PRECEDED WITH THE LIBRARY ROOT NAME $=org.ecemgroup.sharevitalsigns AND KEYS ARE PRECEDED WITH THE LIBRARY NAME #=ShareVitalSigns

Vital Sign	Units	Coding	Action name for Intent filter	Key for vital sign value	Key for vital sign confidence
Heart rate	Beats/min	1	$.MEASURE.HR	#Value_HR	#Confidence_HR
Respiratory rate	Breaths/min	2	$.MEASURE.RR	#Value_RR	#Confidence_RR
Blood oxygen saturation	%	4	$.MEASURE.SPO2	#Value_SPO2	#Confidence_SPO2
Body temperature	°C	8	$.MEASURE.TEMP	#Value_TEMP	#Confidence_TEMP
Systolic blood pressure	mmHg	16	$.MEASURE.BP_SYS	#Value_BP_SYS	#Confidence_BP_SYS
Diastolic blood pressure	mmHg	32	$.MEASURE.BP_DIA	#Value_BP_DIA	#Confidence_BP_DIA
Blood pressure		16|32	$.MEASURE.BP		
Pulse oximetry		1|4	$.MEASURE.PO		

cations to broadcast a "seal of quality", guaranteeing a tested and validated measurement procedure for the returned vital sign.

The ShareVitalSigns library is currently limited to Android OS. Other common OSs offer similar functionality to intents e.g. *UIPasteboard* or *UIActivityViewController* on iOS. However, porting ShareVitalSigns to other platforms is currently not planned.

The ShareVitalSigns work flow is limited to launch the intent from the host application requesting the vital sign. We are currently working on expanding this to the reverse work flow, where a vital sign is measured first and then shared with a processing application. As the vital signs measured are only being shared on the same device and not stored in intermediate physical storage, there are low concerns for privacy and security breaches. As users can choose which measuring application can be installed and no user information is exchanged between application, there is also low risk of identity theft.

V. CONCLUSIONS

The ShareVitalSigns library enables Android applications to request and transfer vital signs between each other. This facilitates the integration of measurement applications into diagnostic and other health monitoring applications. The number of steps for acquiring patient vital signs is reduced. Also, the standardized open source library could increase adoption and consequently enlarge access to third party health measurement applications and devices.

ACKNOWLEDGMENT

REFERENCES

[1] J. M. Ansermino, "Universal access to essential vital signs monitoring," *Anesthesia and analgesia*, vol. 117, no. 4, pp. 883–90, Oct. 2013. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/24023026

[2] Labstyle Innovations, "Dario Glucose meter," 2013. [Online]. Available: http://mydario.com/

[3] Withthings, "Blood Pressure Monitor," 2013. [Online]. Available: http://www.withings.com/bloodpressuremonitor

[4] W. Karlen, G. A. Dumont, C. Petersen, J. Gow, J. Lim, J. Sleiman, and J. M. Ansermino, "Human-centered Phone Oximeter Interface Design for the Operating Room," in *Proceedings of the International Conference on Health Informatics*, V. Traver, A. Fred, J. Filipe, and H. Gamboa, Eds. Rome, Italy: SciTePress - Science and and Technology Publications, 2011, pp. 433–8.

[5] W. Karlen, J. Lim, J. M. Ansermino, G. A. Dumont, and C. Scheffer, "Design Challenges for Camera Oximetry on a Mobile Phone," in *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2012, pp. 2448–51.

[6] D. D. Luxton, R. a. Kayl, and M. C. Mishkind, "mHealth data security: the need for HIPAA-compliant standardization," *Telemedicine journal and e-health : the official journal of the American Telemedicine Association*, vol. 18, no. 4, pp. 284–8, May 2012.

[7] D. Estrin and I. Sim, "Health care delivery. Open mHealth architecture: an engine for health care innovation." *Science*, vol. 330, no. 6005, pp. 759–60, Nov. 2010.

[8] Android 4.4 R1, "Intent — Android Developers," 2013. [Online]. Available: http://developer.android.com/reference/android/content/Intent.html

[9] W. Karlen, H. Gan, M. Chiu, D. Dunsmuir, G. Zhou, G. A. Dumont, and J. M. Ansermino, "Improving the Accuracy and Efficiency of Respiratory Rate Measurements in Children Using Mobile Devices." *PLOS ONE*, vol. 9, no. 6, p. e99266, 2014.

[10] W. Karlen and C. Scheffer, "Design of an Interactive Medical Guideline Application for Community Health Workers," in *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2014, accepted.

[11] WHO, "Integrated Management of Childhood Illness," Geneva, CH, 2008.

[12] D. K. Vawdrey, R. M. Gardner, R. S. Evans, J. F. Orme, T. P. Clemmer, L. Greenway, and F. A. Drews, "Assessing data quality in manual entry of ventilator settings," *Journal of the American Medical Informatics Association : JAMIA*, vol. 14, no. 3, pp. 295–303, 2007.