# Towards a Garment OS: Supporting Application Development for Smart Garments

**Stefan Schneegass**
University of Stuttgart
Stuttgart, Germany
stefan.schneegass@vis.uni-stuttgart.de

**Tobias Birmili**
University of Stuttgart
Stuttgart, Germany
tobiasbirmili@gmail.com

**Mariam Hassib**
University of Stuttgart
Stuttgart, Germany
mariam.hassib@vis.uni-stuttgart.de

**Niels Henze**
University of Stuttgart
Stuttgart, Germany
niels.henze@vis.uni-stuttgart.de

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the Owner/Author.
Copyright is held by the owner/author(s).
*ISWC'14* Adjunct, September 13 – 17, 2014, Seattle, WA, USA
ACM 978-1-4503-3048-0/14/09.

## Abstract
Wearable devices and the development of smart garments emerged into a significant research domain over the last decades. Despite the increasing commercial interest, however, smart garments are almost exclusively developed in academia and the developed systems do not exceed a prototypical level. We argue that the main reason why smart garments cannot be produced on commercially relevant scale today is that they each focus on a specific use case. There is no tool support for application developers and no defined APIs within the software and hardware stack that allows developing useful smart garment applications. In this paper we present our work towards Garment OS, a layered software stack that encapsulates different levels of abstraction. We highlight the design of that system which is based on open web protocols. We present an evaluation with software engineers and derive directions for future work.

## Author Keywords
Smart Garments, Applications

## ACM Classification Keywords
H.5.2 [Information interfaces and presentation]: User Interfaces – misc..

## Introduction

Previous work developed a large variety of prototypes that integrate technology into garments. Researchers developed diverse prototypes including, for example, a sensor jacket for context awareness [6], capacitive neck bands to detect food intake [3] and even proposed to attach textile pressure sensors to the users cheek [5]. Research not only looked at the technical level but also interaction aspects and considered input (e.g., gestures from capacitive sensing [10]) and output (e.g., tactile feedback [9]). A number of successful niche projects have resulted from research prototypes. Despite major research efforts, however, none of the research prototypes has successfully evolved into mass market consumer product. It has been argued that one of the possible reasons is that current smart garment prototypes are made for a single purpose. To overcome this drawback, multipurpose smart garments have been proposed [4]. The vision is to use an application concept similar to application stores known from current smartphones.

Through current application stores for smartphones, such as the Google Play Store[1] or the Apple Appstores[2], users can extend the functionality of their phones and tablets by installing one of the millions of different applications. Application stores are not even limited to mobile devices. Apple's Mac App Store provides applications for desktop computers, Samsung Apps for smart TVs, and Renault's R-link even for cars. Similarly, we believe that the 'app' concept can be an enabler for smart garments. Instead of having cloth for one specific use case it can enable users to download new applications to their garment that extend their possibilities. Applications for smart garments can include sports (e.g., Yoga training supervision), health (e.g., long-term heart rate monitoring [7]), interacting in public spaces (e.g., exploit the garment sensors for interaction with public displays [1]), or social media (e.g., automatically attaches the user's physiological condition to their social media posts). One of the main challenges to this approach is that the processing of data from smart garments is a complex task. Most developers are not experts in signal processing or activity recognition but want to realize their application concepts using a high level API.

[1]http://play.google.com
[2]http://www.apple.com

In this paper, we propose a system that supports developers to build applications for smart garments. It enables them to realize their applications using APIs that provide different levels of abstraction. The APIs provide access to the device specific raw sensor data but also to already processed and device independent information on a more abstract level.

## Supporting Application Developer

Toolkits and frameworks support application developers in a variety of contexts by easing up especially complex development tasks. The most popular ones are used for application or web development. For instance the Google Web Toolkit[3] or the Java Abstract Window Toolkit[4] are popular examples. However, they rather focus on one specific aspect of applications, in this case developing the interface rather supporting the whole development process.

In research, a number of toolkits for processing sensor data have been proposed. The Context Toolkit, for example, allows processing of sensor data and detecting

[3]Google Web Toolkit: http://code.google.com/webtoolkit
[4]Java Abstract Window Toolkit: http://docs.oracle.com/javase/7/docs/api/java/awt/package-summary.html

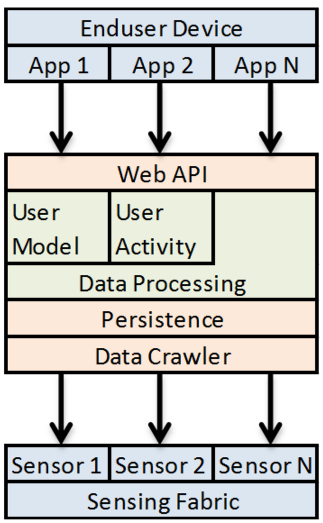


**Figure 2:** Different layers of the Garment OS. The Applications on top access the Garment OS whereas the Sensors at the bottom provide input data.

the context out of them [11]. The CRN Toolkit is a high level framework for implementing activity and context sensitive applications using a graphical configuration editor [2]. The toolkit shows its usefulness in a couple of projects. Sahami et al. proposed a toolkit that allows accessing sensors on remote phones (e.g., the camera to obtain remote images) [13]. Focusing more on the developer than on obtaining sensor data, Schneegass and Alt provide a system that eases up the sensor integration on public displays [12]. Such toolkits provide important functionality and ease the development process up.

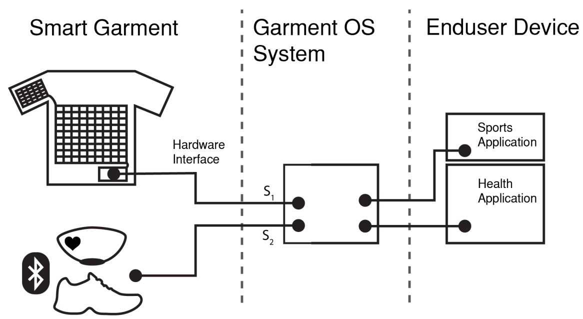


**Figure 1:** Overview of the Garment OS architecture.

## Architecture of a Garment OS

Supporting developers in building a very wide range of applications for smart garments, we envision a *Garment OS* that works between the sensing fabric (i.e., the hardware) and the applications (cf., Figure 1). These applications could run either on a new off the shelf device or the user's mobile phone. We propose the *Garment OS* as a layered system consisting of different layers (cf., Figure 2). The lowest layer (i.e., Data Crawler) is connected to the fabric sensors (e.g., Capacitive Sensors [3]) and actuators (e.g., Electronic Muscle Stimulation feedback [8]). Different applications can use the Web API (i.e., the highest layer) to access the sensor information concurrently.

The Data Processing layer provides the raw sensor data. In addition, developers can use the preprocessed sensor data and add custom or pre-defined filters to the raw sensor data. Most commonly used algorithms are pre-defined and can easily be applied to the sensor data. This functionality is especially important for detecting non-standard gestures or postures.

In addition to these values, we envision two specific components that support the developer. First, a User Model is generated out of the sensor data. This model contains, on the one hand, the posture of the user similar to the Microsoft Kinect skeleton and, on the other hand, information about the physiological state of the user (e.g., heart rate). This especially eases the development of applications since the developer can access pre-defined postures and gestures (e.g., user waving the hand) and does not need to deal with the raw sensor data. Additionally, the system keeps track of the physiological state of the user and the developers can receive events in their application that react to this state – for instance, a music player that plays calming music if the user is nervous. Second, an activity recognition component detects basic activities (e.g., running). Thus, application can be triggered by a specific activity performed by the user and, therefore, assist the user automatically. These components are especially important considering the fact that applications should run on different smart garments with different sensors. All the sensors are combined in the Garment OS and serve as input to determine the User Model and the User Activity.

The data processing layer creates events that the developer can listen to in the application. For instance, the application receives an event as soon as the activity of the user changes to running or the physiological value in the user model exceeds a specific level that indicates an increased level of stress.

One of the main goals of this approach is to allow developers the creation of applications that are independent from the actual garment. Since we envision that different smart garments with different sensors will be available, the garment OS needs to chose appropriate sensors from the sensors available in the garment. In addition, this layer deals with different artifacts (e.g., from sensor displacement or the user‘s motion). These artifacts should be compensated automatically by the garment OS.

## Prototype

We implemented a prototype of the *Garment OS*. We thereby focus on web technologies to enable an easy interoperability between different devices with different operating systems and programming languages. The system itself consists of five layers, namely, sensors, data crawler, persistence, data processing, and Web API layer.

*Input Sensors*

The sensor layer encapsulates the access to different physical sensors. We implemented three different sensor wrappers to connect sensors to our system. As a first example, we implemented a mobile phone application that is on the one hand providing the accelerometer data as well as the possibility to provide user-specific labels. Second, we developed a wrapper that connects the BioHarness 3[5] to our system. Thereby, it sends the physiological data (i.e., heart rate, breathing rate, and acceleration) to the system.

*Data Crawler*

The data crawler is the interface between the sensors (wrappers) and the persistence layer. A Tornado server[6] asynchronously accepts the sensor data and forwards it to the persistence layer.

*Persistence*

The persistence layer is based on a key-value storage. In our implementation we use Mongo DB that stores all the received information. Application developers can store recorded sensor data in the database that will assist them in the development process.

*Data Processing*

Since developers should not deal with the data processing, we implemented a filter chain that can be applied to the sensor data. Thereby, a number of basic filters (e.g., average of the last X seconds) can be combined to form a chain. A combination of filters can for instance detect whether a user is jumping or not. These chains can be used by application developers in contrast to using the raw data. In the current version of the system, we did not implement the user model as well as the activity recognition algorithms.

*Web API*

The Web API uses Web Sockets to create connections between applications and the Garment OS (cf., Figure 3). By using Web Sockets, we allow the connection of smartphones as well as other mobile devices such as smart watches.

[5] http://zephyranywhere.com/products/bioharness-3/

[6] http://www.tornadoweb.org/

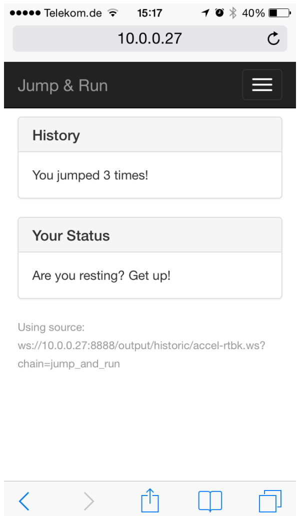


**Figure 5:** One of the applications developed by the participants during the evaluation study.

**Figure 6:** Participants during the evaluation study testing the application.

```
socket = new WebSocket('ws://sedm.dev:8888/accel-demo-1.ws?chain=jump_and_run');

socket.onmessage = function(message){
      data = JSON.parse(message.data);
      if(data.is_jumping) {
            $('#jump').show();
            $('#run').hide();
      } else if(data.is_running){
            $('#run').show();
            $('#jump').hide();
      }
}
```

**Figure 3:** By using Web Sockets the data can be accessed easily with a wide variety of devices.

*Garment Interface*
In addition, we implemented a Garment Interface that allows the developer to record and reload sensor data to test the application. At the same time it is possible to visualize the data (cf., Figure 4). This is especially helpful for debugging developed applications.

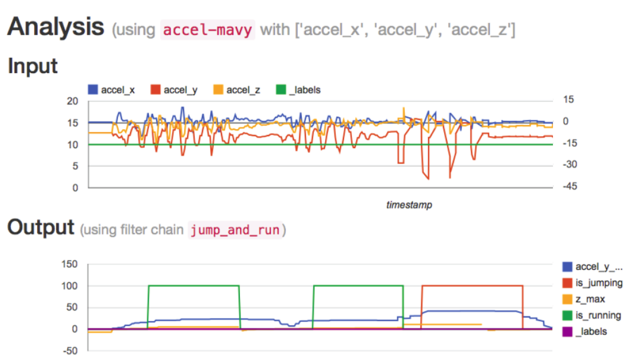


**Figure 4:** The garment interface allows the developer to review recorded data and the effect of applied filters and filter chains.

## Evaluation
To evaluate the developed system, we invited six potential developers of smart garment applications (all male) aged 24 to 26 ($M = 25.00$, $SD = 0.63$). All participants worked as full or half-time software developers.

*Procedure*
After arriving at the lab, we first introduced the system to the participants. We explained the most important functions and provided examples of how to use the system. Then, participants had to perform an actual development task. We asked them to develop a simple application that determines if a user is running or jumping and counts how often a user is jumping. After the participants performed the task, we asked them to fill in a questionnaire and we collected qualitative feedback.

*Results*
In total, all participants were able to fulfill the proposed task and each of them developed an application that counts how often the user is jumping (cf., Figure 5 and Figure 6). Therefore, they used the accelerometer data that is either provided by the bioharness 3 or the smartphone sensor.

In the questionnaire (5 point likert scale), participants stated that the system especially supported the developer during the recording ($Mdn = 4$) and analyzing ($Mdn = 3.5$) of the sensor data. Furthermore, participants are convinced that the system is flexible enough to allow a variety of different applications ($Mdn = 4$). In particular, participants appreciated the support for implementing an algorithm for activity recognition. One participant stated, for example, "The filter chains clearly reduced the complexity of the processing of the movement data." A clear direction for further improvement is the system's GUI. A participant stated, for example, that foremost, the GUI needs to be improved for productive use."

## Discussion and Conclusion

In this paper, we present our concept of a Garment OS that aims to support developers to build applications for smart garments. The Garment OS separates the applications and the garments from each other and, thus, allows using multiple applications with different (smart) garments. We developed a web-based prototype including three prototypical sensors. Furthermore, we conducted an evaluation study in which six developers developed an application for the Garment OS. The results show that developers are able to use the Garment OS and that the system eases the development of applications for smart garments.

Nevertheless, there are a couple of open challenges that need to be addressed. From the application developer's perspective, it needs to be investigated to which extent the data can be preprocessed without loosing flexibility in the application. Since physiological data gathered through smart garments are highly personal, the data access needs to be realized in a privacy preserving way. Furthermore, appropriate input and output modalities for smart garments need to be investigated with respect to being applicable to arbitrary garment applications.

## Acknowledgments

The research leading to these results has received funding from the European Union 7th Framework Programme ([FP7/2007-2013]) under grant agreement no. 323849.

## References

[1] Alt, F., Schneegass, S., Schmidt, A., Müller, J., and Memarovic, N. How to evaluate public displays. In *Proc. of PerDis*, ACM (New York, NY, USA, 2012).

[2] Bannach, D., Lukowicz, P., and Amft, O. Rapid prototyping of activity recognition applications. *Pervasive Computing, IEEE 7* (2008), 22–31.

[3] Cheng, J., Amft, O., Bahle, G., and Lukowicz, P. Designing sensitive wearable capacitive sensors for activity recognition. *Sensors Journal, IEEE 13* (2013), 3935–3947.

[4] Cheng, J., Lukowicz, P., Henze, N., Schmidt, A., Amft, O., Salvatore, G. A., and Troster, G. Smart textiles: From niche to mainstream. *IEEE Pervasive Computing 12*, 3 (2013), 81–84.

[5] Cheng, J., Okoso, A., Kunze, K., Henze, N., Schmidt, A., Lukowicz, P., and Kise, K. On the tip of my tongue: a non-invasive pressure-based tongue interface. In *Proc. Augmented Human*, ACM (2014), 12.

[6] Farringdon, J., Moore, A. J., Tilbury, N., Church, J., and Biemond, P. D. Wearable sensor badge and sensor jacket for context awareness. In *Proc. ISWC*, IEEE (1999), 107–113.

[7] Pacelli, M., Loriga, G., Taccini, N., and Paradiso, R. Sensing fabrics for monitoring physiological and biomechanical variables: E-textile solutions. In *Proc. IEEE-EMBS* (2006), 1–4.

[8] Pfeiffer, M., Schneegass, S., Alt, F., and Rohs, M. A design space for electrical muscle stimulation feedback for free-hand interaction. In *Proc. CHI Workshop on Assistive Augmentation.* (2014).

[9] Pfeiffer, M., Schneegass, S., Alt, F., and Rohs, M. Let me grab this: A comparison of ems and vibration for haptic feedback in free-hand interaction. In *Proc. Augmented Human* (2014).

[10] Rekimoto, J. Gesturewrist and gesturepad: Unobtrusive wearable interaction devices. In *Proc. of ISWC* (2001), 21–27.

[11] Salber, D., Dey, A. K., and Abowd, G. D. The context toolkit: Aiding the development of context-enabled applications. In *Proc. CHI* (1999).

[12] Schneegass, S., and Alt, F. Senscreen - a toolkit for supporting sensor-enabled multi-display networks. In *Proc. PerDis*, PerDis '14 (2014), 92–97.

[13] Shirazi, A. S., Winkler, C., and Schmidt, A. Sense-sation: An extensible platform for integration of phones into the web. In *Internet of Things*, IEEE (2010), 1–8.