Accessing Mobile IoT:
Application Development With Cordova Accessor Host

Ilana Levy   University of California, Los Angeles (UCLA), Los Angeles, CA, USA
Dr. Anne Ngu   Texas State University, San Marcos, TX, USA

Motivation

Internet of Things (IoT)

IoT consists of smart devices collecting sensor data for user applications.

➔ IoT platforms (Google, Microsoft, etc) use different standards and languages
➔ Difficult for "Things" from different platforms to exchange information.

What Are Accessors

Accessors act as intermediaries which allow smart "Things" to communicate and share data.

➔ Each accessor has input and output ports to exchange information
➔ Manage/process the data transfer between "Things", servers, UI, etc
➔ Easily interchangeable, just change the connections

Why Cordova Host[2]

Accessors run on an accessor host platform similarly to how websites run on a browser. The host controls most of the accessor's functionality.

➔ Cordova is the only Host lightweight enough to run on mobile phones
➔ Cross platform capabilities (Android, IOS, browser) minimizes repetition
➔ Preliminary Cordova Host developed by the TerraSwarm organization at UC Berkeley is low level and unintuitive
➔ There has been little development of real world applications to demonstrate its potential over traditional IoT platforms

Fall Detection Method

Gather Data:

■ I used the Microsoft Band 2 as a smart wrist wearable to gather accelerometer data and send it via bluetooth to a smartphone accessor.
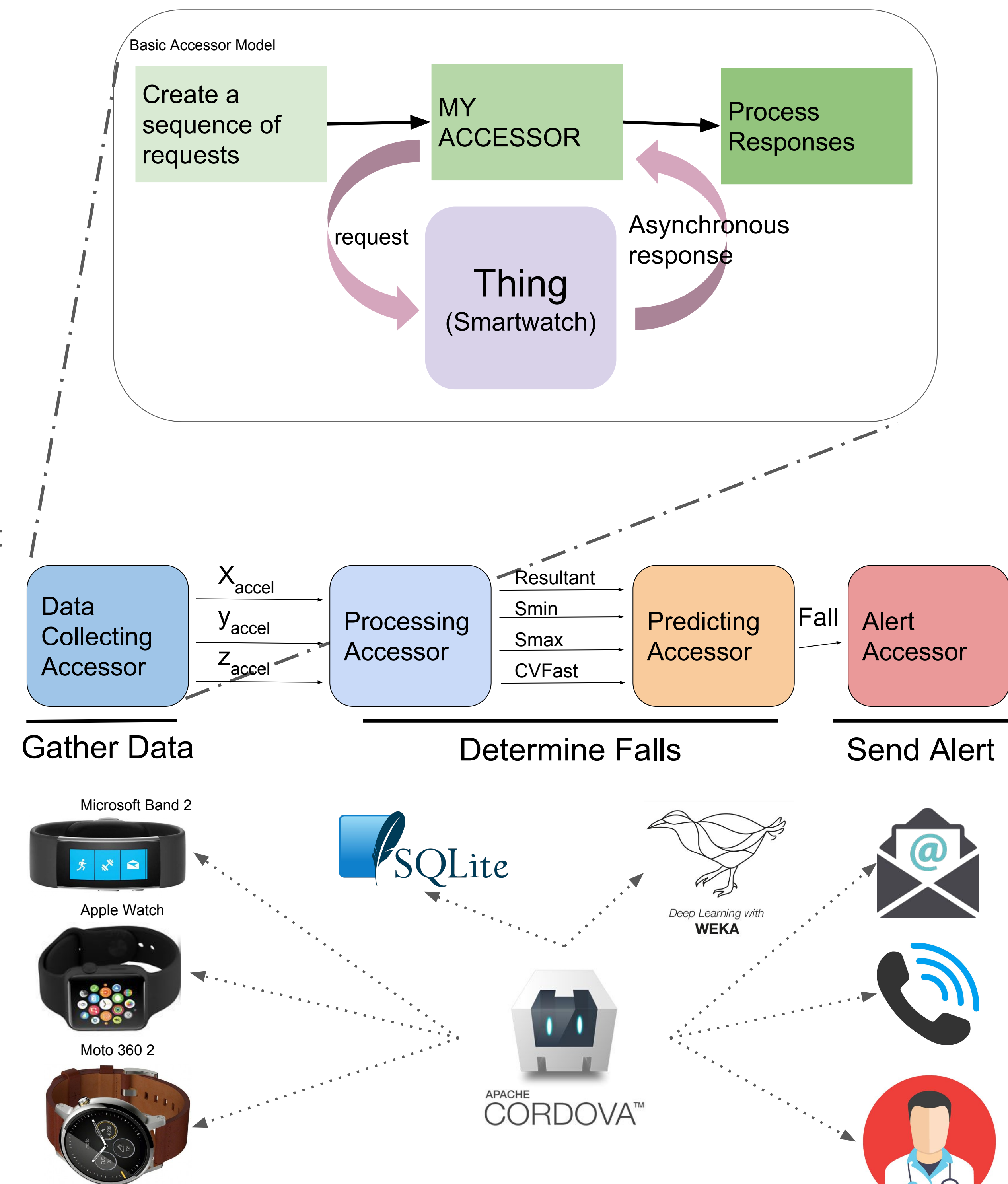
Determine Falls:

■ A processing accessor downsamples the acceleration data and stores it in a SQLite table on the phone and extracts the important information for fall detection.

■ The predicting accessor takes the fall values from the processing accessor and uses a machine learning model to detect if the data indicates a fall. I use a Naive Bayes predicting model from Weka. If a true fall is detected, the accessor fires an output to the alert accessor.

Send Alert:

■ Upon receiving input, the accessor notifies contacts about the fall

Abstract

Cordova is the sole model for mobile Accessor technology, but there has been almost no development of real world applications on a Cordova Host. To demonstrate the benefits of a Cordova Accessor Platform over traditional IoT models, I created an android Fall Detection application which streamlines accelerometer data from a Microsoft Band 2 to a mobile phone.

Basic Accessor Model

Create a sequence of requests → MY ACCESSOR → Process Responses

request / Asynchronous response — Thing (Smartwatch)

Data Collecting Accessor → x_{accel}, y_{accel}, z_{accel} → Processing Accessor → Resultant, Smin, Smax, CVFast → Predicting Accessor → Fall → Alert Accessor

Gather Data   Determine Falls   Send Alert

Microsoft Band 2 — Apple Watch — Moto 360 2 — SQLite — Deep Learning with WEKA — APACHE CORDOVA™

References

1. Ngu, Anne & Wu, Yeahuay & Zare, Habil & Polican, Andrew & Yarbrough, Brock & Yao, Lina. (2017). Fall Detection Using Smartwatch Sensor Data with Accessor Architecture. 81-93. 10.1007/978-3-319-67964-8_8.
2. Brooks, Christopher & Jerad, Chadlia & Kim, Hokeun & Lee, Edward A. & Lohstroh, Marten & Nouvellet, Victor & Osyk, Beth & Weber, Matt (2018). A Component Architecture for the Internet of Things. *Proceedings of the IEEE (Volume: PP, Issue: 99)*

Conclusion

Cordova has proven to be a valuable host environment for mobile accessor technology.

➔ The accessors running on a mobile processor successfully handled a continuous data stream without crashing
➔ Exchangeability of accessors in this application demonstrates how it can be updated more efficiently than current IoT systems
➔ Replace outdated accessors independently, without replacing or updating the entire system
➔ This technology can be modified for numerous systems and entirely new applications by swapping out accessors

Easy Accessor Modifications:

■ Data Collecting Accessor:
  ○ Replace MSBand with Apple Watch, Moto 360, Samsung, etc
  ○ Collect and use additional data (eg Heart Rate, blood pressure)

■ Processing Accessor:
  ○ Change sampling rate, update storage method, etc

■ Predicting Accessor:
  ○ Use different models for new predictions and applications, such as analyze gait from accelerometer data to detect early Alzheimer's

■ Alert Accessor:
  ○ Could be modified to also alert doctor, 911, or other services

Future Work

Accessors have the potential to make large-scale smart systems such as smart cities a reality. Their ability to communicate with multiple smart "Things" across multiple platforms and their adjustability sets them apart from other IoT systems. Future work includes:

Energy Consumption

As accessors are triggered by Discrete Events there is evidence to suggest that they consume less energy and require less storage space than traditional IoT systems. More research should be conducted to validate these claims as efficiency is a key element in modern tech.

Easier Plugin Development

An obstacle in Cordova is developing plugins which allow the JavaScript code to interact with native code (Java, C, etc). Simplifying this process will enable Cordova to serve a broader user base.