APISENSE®: Mobile crowd-sensing made easy!
Romain Rouvoy

To cite this version:
Romain Rouvoy. APISENSE®: Mobile crowd-sensing made easy!. Keynote of ECAAS’16. 2016. <hal-01332594>

HAL Id: hal-01332594
https://hal.inria.fr/hal-01332594
Submitted on 13 Oct 2016
Spirals
Self-adaptation for distributed services and large software systems

Presentation

Spirals is conducting research activities in the domains of distributed systems and software engineering.

Spirals aims at introducing more automation in the adaptation mechanisms of software systems, in particular, transitioning from adaptive systems to self-adaptive systems. Spirals targets especially two properties: self-healing and self-optimization. With self-healing, Spirals aims at studying and tailoring data mining and machine learning solutions for the design and implementation of software systems. This contributes to the goal of obtaining solutions for automatic software repair. With self-optimization, Spirals aims at sharing, collecting, and analyzing distributed behaviors and data to continuously tailor, optimize, and keep under working conditions software systems. This participates to the goal of obtaining eternal distributed systems.

Spirals is a joint project-team between Inria and the University of Lille – Sciences and Technologies within UMR 9189 CRISAL. Spirals originates from the ADAM project-team (2008-13).

Available Positions

- PhD & Internship
- Privacy-aware data dissemination in mobile networks
- Cartography of the Quality of Experience for Mobile Internet Access
- R&D Engineer
- Développeur applications mobiles pour plateforme de géolocalisation indoor
- Research Software Engineer in Automatic Repair

Links

- Software Engineering seminar
- SomaInria associated team
- Videos

Spirals Twitter

Tweets by @SpiralsTeam
Welcome to CRAWDAD

CRAWDAD is the Community Resource for Archiving Wireless Data At Dartmouth, a wireless network data resource for the research community. This archive has the capacity to store wireless trace data from many contributing locations, and staff to develop better tools for collecting, anonymizing, and analyzing the data. We work with community leaders to ensure that the archive meets the needs of the research community.

CRAWDAD is grateful to its current and past sponsors.

Latest News

new version of CRAWDAD dataset - factory channel gain measurements - June 13, 2016

A new version of the init/factory dataset has been added to CRAWDAD.

init/factory

Contributed by Dimitri Block, Niels Hendrik Fliedner, Uwe Meier.

Measurement of the channel gain for multiple distances within a factory environment. There are two new trace sets in this version.

If you do use these data, please let us know, and you can use the DOI 10.15783/C76S3K to do so. BibTeX and BIS are provided on the website.
Smartphones, mobile PCs, tablets and mobile routers with cellular connection

- **300 million** mobile PCs, tablets and mobile router subscriptions
- **1.9 billion** smartphone subscriptions

---

**5.6 BILLION** smartphone subscriptions by the end of 2019

- Orange: Mobile PCs, tablets and mobile router subscriptions
- Red: Smartphone subscriptions
Crowd & sensing
Crowd Sensing | kraʊd:sɛnsɪŋ |

«Capability of lifting a (large) diffuse group of participants to delegate the task of retrieving trustable data from the field. This includes:

• Participatory sensing involves the user in the sensing task (eg. surveys)
• Opportunistic sensing uses mobile sensors carried by the user (eg. Smartphones)»
Applications to data visualisation

source: http://opensignal.com
Applications to IoT monitoring
Applications to crowdsourcing

source: http://fr.clicandwalk.com
Xperium: Mobility Analysis

97.8 kg eq. CO₂
0.12 kg eq. CO₂ per km

Itinerary visualisation

Transport classification

Android app
for collecting data
# PRACTIC : Human Analysis

**PRACTIC jeu-concours (march 10th - april 21th 2014)**

## All Campaigns (except Caen)

<table>
<thead>
<tr>
<th>Category</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>112,371 of data</td>
<td>Or 39 days of data per user (average)</td>
</tr>
<tr>
<td>97 volunteers</td>
<td>Of which:</td>
</tr>
<tr>
<td></td>
<td>- 64% male</td>
</tr>
<tr>
<td></td>
<td>- 80% age under 30</td>
</tr>
<tr>
<td></td>
<td>- 73% single</td>
</tr>
<tr>
<td></td>
<td>- 35% income &lt; 1500€/m.</td>
</tr>
<tr>
<td>68% of students</td>
<td>Of which:</td>
</tr>
<tr>
<td></td>
<td>- 43% Computer Sc./ingeneers</td>
</tr>
<tr>
<td></td>
<td>- 22% Communication</td>
</tr>
<tr>
<td>14 device brands</td>
<td>Of which:</td>
</tr>
<tr>
<td></td>
<td>- 41% Samsung</td>
</tr>
<tr>
<td></td>
<td>- 24% LGE</td>
</tr>
<tr>
<td></td>
<td>- 12% Sony</td>
</tr>
<tr>
<td>48 device models</td>
<td>Of which:</td>
</tr>
<tr>
<td></td>
<td>- 12% LGE Nexus 5</td>
</tr>
<tr>
<td>13 telecom operators</td>
<td>Of which:</td>
</tr>
<tr>
<td></td>
<td>- 77% Free, Orange, Bouygues and SFR;</td>
</tr>
<tr>
<td></td>
<td>- 12% unknown</td>
</tr>
<tr>
<td>14 Android versions</td>
<td>Of which:</td>
</tr>
<tr>
<td></td>
<td>- 26% in 4.4.2</td>
</tr>
</tbody>
</table>
Frequency and duration of sessions on a smartphone and a tablet (occurrences of the number of sessions according to 3 levels of duration)

**Jeu de données d’un smartphone**
- Première donnée le 19 avril 2013
- Avril : 2371 sessions sur 12 jours
- Mai : 3579 sessions sur 25 jours
- Juin : 1000 sessions sur 4 jours
- Juillet : 7090 sessions sur 31 jours
- Août : 5507 sessions sur 29 jours
- Septembre : 8304 sessions sur 30 jours
- Octobre : 8188 sessions sur 31 jours
- Novembre : 6287 sessions sur 30 jours
- Décembre : 4867 sessions sur 26 jours
- Soit une présence de 218 jours sur 257 (84.82%) pour 47193 sessions
- Moyenne par jour : 216.5 sessions
- Écart-type par jour : 89.03

**Jeu de données d’une tablette**
- Première donnée le 11 avril 2013
- Avril : 561 sessions sur 16 jours
- Mai : 824 sessions sur 29 jours
- Juin : 513 sessions sur 27 jours
- Juillet : 688 sessions sur 25 jours
- Août : 449 sessions sur 24 jours
- Septembre : 387 sessions sur 29 jours
- Octobre : 288 sessions sur 27 jours
- Novembre : 338 sessions sur 21 jours
- Décembre : 218 sessions sur 26 jours
- Soit une présence de 224 jours sur 265 (84.5%) pour 4266 sessions
- Moyenne par jour : 19 sessions
- Écart-type par jour : 17.34
How does it work?
How does it work?

1. Describe

2. Collect

Privacy & Energy
How does it work?

1. Describe
2. Collect
3. Make sense!

Privacy & Energy

- Open data
- Applications
- Visualizations
- Notifications
- Studies
Bee makes it easy to collect data with crowds of mobile phone sensors.

No mission here :-(

Click the Bee store button to add some!

A crowdsensing solution.

Resources:
APISENSE Bee

Romain Rouvoy

Details  Ratings and Reviews  Related

SCREENSHOTS

Get

This app is designed for both iPhone and iPad

Rating: 4*

LINKS
Developer Website

© 2013 APISENSE®
Collect data easily with crowds of mobile phone sensors. Make sense and innovate on top of real world data feedback, in real time!
**BottleNet**

**Crop panel**

- Test your network connection

**Control panel**

- **Check**
  - Stop and disable the crop on clients. They won’t be able to start or subscribe anymore.

**Data panel**

- **Participants**: 4
- **Synchronizations**: 178
- **Collected data**: 217 KB
- **Last upload**: 28/05/2016 - 13:55

**QRCode**

The QRCode generated represents the crop’s identifier. It can be used from the Bee application to install unlisted crops or manual installation.

Embed it on your own website:

```html
<img src="apisense.io/crop/oPNPvJbwJ55v9qBGZJ1/qrcode" alt="QRCode to access crop BottleNet"/>
```

© 2015 APISENSE®  Terms
```javascript
var recorder = require('recorder');
var gps = require('location');
var battery = require('battery');
gps.onLocationChanged({mode: gps.PASSIVE, distance: 100}, function()
  recorder.save({
    'latitude': gps.latitude(),
    'longitude': gps.longitude(),
    'speed': gps.speed(),
    'accuracy': gps.accuracy()
  });
battery.onStateChanged(function(data) {
  recorder.sync();
});
```
```javascript
rest.prepareFilter("days", function(data){
    var result = [];
    var date;
    for (var ele in data) {
        date = new Date(ele.metadata.timestamp);
        date.setHours(0);
        date.setMinutes(0);
        date.setSeconds(0);
        date.setMilliseconds(0);
        if (result.indexOf(date.toString()) == -1) {
            result.push(date.toString());
        }
    }
    return result;
});

rest.prepareFilter("byDay", function(data){
    var result = {};
    var date;
    for (var ele in data) {
        date = new Date(ele.metadata.timestamp);
        date.setHours(0);
        date.setMinutes(0);
        date.setSeconds(0);
        date.setMilliseconds(0);
        if (result[date] === undefined) {
            result[date] = [];
        }
        result[date].push(ele.body);
    }
    return result;
});
```
Device-level Sensing Task

**Event**
- onLocationChange
- onNetworkStateChange
- onBatteryStateChange
- onPhoneCall

**Action**
- Save Data
- Capture Media
- Survey
- Notify

**Facade**
- Location
- User Interaction
- Device State
- Battery Level

**Actions**
- subscribe
- act
- publish
var location = requires('location');
var trace = requires('honeycomb');
var telephony = requires('gsm');

location.onLocationChange(function(event){
    trace.sync({
        lat : event.latitude,
        lng : event.longitude,
        signal : telephony.signalStrength()
    });
});
### Crowd-scale Sensing Jobs

<table>
<thead>
<tr>
<th>function</th>
<th>code</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>sense</code></td>
<td><code>sense(function() { ... })</code></td>
</tr>
<tr>
<td><code>accept</code></td>
<td><code>accept(function() {</code></td>
</tr>
<tr>
<td></td>
<td><code>if (network.connectionType() == 'mobile')</code></td>
</tr>
<tr>
<td></td>
<td><code>return {battery: battery.level()};</code></td>
</tr>
<tr>
<td><code>recruit</code></td>
<td><code>};</code></td>
</tr>
<tr>
<td><code>ranking</code></td>
<td><code>ranking(function(users) {</code></td>
</tr>
<tr>
<td></td>
<td><code>return users.sort('battery');</code></td>
</tr>
<tr>
<td><code>geoCoverage</code></td>
<td><code>geoCoverage([[[50.614291,3.13282],[50.604159,3.15239]], '500 m']);</code></td>
</tr>
<tr>
<td><code>timeCoverage</code></td>
<td><code>timeCoverage('30 min','1 H');</code></td>
</tr>
<tr>
<td><code>duplicate</code></td>
<td><code>duplicate(1);</code></td>
</tr>
</tbody>
</table>
Crowd-scale Sensing Job

1. Virtual sensor deployment [Chowdhury10]

Crowd-scale Sensing Job

1. Virtual sensor deployment [Chowdhury10]
2. Connecting to physical devices

[Chowdhury10] **A survey of network virtualization.** *Computer Networks.* 2010
Crowd-scale Sensing Job

1. Virtual sensor deployment [Chowdhury10]
2. Connecting to physical devices
3. Assigning sensing tasks

Crowd-scale Sensing Job

1. Virtual sensor deployment [Chowdhury10]
2. Connecting to physical devices
3. Assigning sensing tasks
4. Executing sensing tasks

Evaluation of APISENSE®

**REDUCTION OF 50–80% OF DATA EXCHANGES**

**UP TO 82% ENERGY SAVINGS**

**MAXIMUM COVERAGE LOSS OF 2%**

<table>
<thead>
<tr>
<th></th>
<th>W/B-Scanner opportunist</th>
<th>Citizen journalist participative</th>
</tr>
</thead>
<tbody>
<tr>
<td>APISENSE®</td>
<td>4</td>
<td>9</td>
</tr>
<tr>
<td>Anonysense</td>
<td>5</td>
<td>N/A</td>
</tr>
<tr>
<td>Pogo</td>
<td>4</td>
<td>N/A</td>
</tr>
<tr>
<td>MyExperience</td>
<td>N/A</td>
<td>27</td>
</tr>
<tr>
<td>Medusa</td>
<td>N/A</td>
<td>45</td>
</tr>
<tr>
<td>PRISM</td>
<td>??</td>
<td>330</td>
</tr>
</tbody>
</table>
Mobile Data Science Platform

Simplifying mobile, human and environmental data analytics.

GET STARTED
Book chapters


Conferences


Workshops


Dissemination

Collecting exception in the wild

Warning log Taxonomy

Error log Taxonomy
Assessing Machine Learning Models

• User context recognition implementation: ~ 30 lines

```javascript
... accelerometer.onChange(function(acc) { buffer.push(acc) });
// Learning phase
dialog.display({ message: "Select movement", spinner: classes },function(pattern){
  accelerometer.onChange(function(acc) { buffer.push(acc) });
  sleep('5s')
  model.record(attributes(buffer), pattern);
  buffer = new Array();
  return;
});
...
// Exploitation phase
time.schedule({ period: '5s' }, function() {
  trace.add(
    { position: model.evaluate(attributes(buffer)),
      stats: model.statistics() });
  buffer = new Array();
});
```

Representative Confusion Matrix

<table>
<thead>
<tr>
<th>Predicted class</th>
<th>Walk</th>
<th>Jog</th>
<th>Stand</th>
<th>Sit</th>
<th>Up</th>
<th>Down</th>
<th>Acc (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Walk</td>
<td>66</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>94.3</td>
</tr>
<tr>
<td>Jog</td>
<td>0</td>
<td>21</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>Stand</td>
<td>4</td>
<td>0</td>
<td>40</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>90.9</td>
</tr>
<tr>
<td>Sit</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>83</td>
<td>0</td>
<td>0</td>
<td>97.6</td>
</tr>
<tr>
<td>Up stair</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>22</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>Down stair</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>11</td>
<td>100</td>
</tr>
</tbody>
</table>

10/13/16
**Incentive**: the model of a free service between *Quantified-self* and *Mydata*.
#include "Sensor.h"
#include "Module.h"
#include "Channel.h"

Channel* mlog = new LogChannel();
Sensor* sensorTmp = new Sensor(A0, "temperature", &convertTemperature);
Sensor* sensorLum = new Sensor(A2, "Lumiere", &convertLumiere);
Module* myModule = new Module();

void setup() {
  Serial.begin(9600);
}

void loop() {
  myModule->load(sensorTmp);
  myModule->load(sensorLum);
  myModule->setChannel(mlog);
  myModule->updateM();
  delay(300000);

  int convertTemperature(int sensorTmpVal){
    float voltage = (sensorTmpVal/1024.0) * 5.0;
    float temperature = (voltage - .5) * 100;
    return temperature;
  }
}