Contributions to the Autonomy of Ubiquitous Software Systems

Romain Rouvoy

Habilitation defense (HDR)

Friday 5th December 2014
A few facts about me

• 2003–06: **Ph.D. thesis** from USTL / INRIA
  – *Supervision*: P. Merle & J.M. Geib
  – *Topic*: Middleware / components / transaction

• 2007–08: **Postdoc researcher** @ Univ. of Oslo
  – *Member of the ND group* (IFI)
  – *Contributions to the EU FP6 MUSIC R&D project*

• 2008–: **Associate professor** @ Univ. of Lille 1
  – *Member of the Spirals team* (Lille 1 & Inria)
  – *Head of the CS Master on software engineering*
  – *Distributed systems & software engineering*
    • Middleware for ubiquitous computing
Evolution of connected devices

Smartphones, mobile PCs, tablets and mobile routers with a cellular connection

Evolution of application usages

How to deal with the generated data traffic?

Ubiquitous software systems

How to control the energy consumption?

How to handle resources’ sporadicity?
Challenges: How to...

- **Program** ubiquitous software systems?
- **Control** ubiquitous software systems?
- **Identify** the relevant control features?
Our approach: To support...

1. Elastic Programming Models
2. Engineering of Feedback Control Loops
3. Multi-dimensional Context Monitoring
2011–2013

Enabling opportunistic code offloading

ELASTIC PROGRAMMING MODELS
Ubiquitous software systems require

• **Discovery** of resources and capabilities
  – Including local and remote discovery
  – Cyber foraging [Satya’01, Goyal’04]

• **Optimisation** of the computing resources
  – Including energy and network consumptions
  – Code offloading [MAUI’10, Cuckoo’12]

• Leverage of interoperability and **serendipity**
  – Exploiting unexpected opportunities

---


**Goyal’04.** S. Goyal et al. A lightweight secure cyber foraging infrastructure for resource-constrained devices. WMCSA’04

**MAUI’10.** E. Cuervo et al. MAUI: making smartphones last longer with code offload. MobiSys’10

Elastic Programming Model

Identifier → Resource → URI

Verbs → Representations


Macchiato ubiquitous middleware

- **Distributed middleware**
  - Based on JavaScript
- **Actor-based model**
  - RDF messages
- **Support for discovery**
  - Local discovery (UPnP)
  - Location-based discovery
- **Support for offloading**
  - Remote actor deployment
  - Deployment optimisation

```javascript
macchiato.create("search-actor").as(
    function (msg, context) {
        context.http.get(
            "http://api.developer.sears.com/v1/products" +
            "?apikey=" + context.searsapikey
            + "&store=Sears&searchType=keyword&keyword="
            + msg.search,
            function (result) {
                context.send(context.parseJSON(result));
            })
    });
```
Enabling elasticity in Macchiato

- Actor graph $\rightarrow$ SAT problem
- Discovery $\rightarrow$ constraints
- Objective function $\Rightarrow$ network exchanges
- Reasoning engine
  - Triggers upon change in the graph
  - Considers sub-optimal solutions
  - Queues pending messages
  - Keeps a local copy of actors
Cumulated energy consumption

Cumulated energy consumption

- Client Only
- Dynamic offloading
- Static offloading

Limitations of the approach

- User-centric process
  - Lack of coordination
- Embedded heuristics
  - Lack of customization
- Cost of heuristics integration
  - Lack of extensibility
- Confidence in heuristics
  - Lack of feedback

How to address control as a crosscutting concern?
2009–2012

*Self-Adaptive very Large disTributed sYstems*

ENGINEERING OF
FEEDBACK CONTROL LOOPS
Engineering of Feedback Control Loops

- **Decision Making**
  - Measures
  - Decisions
  - Achieve externally specified goals

- **Monitoring**
  - System outputs
    - Response time, utilization, throughput
  - Events

- **Reconfiguration**
  - System control inputs
    - Queue size, scheduling policy, concurrency level
  - Actions

- **Target System**
  - Sensors
  - Effectors

- **Control Plan**
  - Base plan
Engineering of Feedback Control Loops

1. Raise the level of abstraction
2. Fine-grained decomposition of FCL elements
3. Explicit interactions
4. Provide reflection capabilities
5. Embed remoting

**Feedback Control Loop**
- Sequence of interconnected processes
- Input \times State \rightarrow Output
- Reactive
- Concurrent
- Dynamic

**Domain-Specific Modeling**
- Abstraction
- Automation
- Analysis

**The Actor Model**
- Message passing actor networks
- Message \times State \rightarrow Message(s)
- Reactive
- Concurrent
- Dynamic
- Scalable
- Remoting through location transparency
FCDL in a nutshell

Adaptive Element

- Actor-like component
- Sensors
- Effectors
- Processors
- Controllers
- Input/output ports & properties
- Active / passive
- Implementation
  - Imperative code (e.g. Java)
  - CEP Rules (e.g. Drools)
  - STM (e.g. bzz/heptagon)
  - Matlab
  - BASH
  - ...
- Interaction contracts

\[ \alpha = \langle \text{self}; \downarrow \text{(input)}; \uparrow \text{(output?)} \rangle \parallel \langle \uparrow \text{(setPeriod)}; \emptyset; \uparrow \text{(period)} \rangle \]

\[ u(t) = u(k-1) + K_e(k) \]
Ubiquitous computing – key results

• **Publications (3 PhD thesis / 8 journals / 15 conferences)**
  – N. Petitprez et al. Connecting your Mobile Shopping Cart to the Internet-of-Things. In: *DAIS’12 *

• **Collaborations**
  – University of Nice (ANR SALTY, P. Collet & F. Krikava)
  – University of Oslo (EGIDE PHC Aurora & SeaS associate team)

• **Grants**
  – ANR ARPEGE SALTY FUI Cappucino
  – FUI Macchiato PIA Datalyse
A Focus on context monitoring

- **A critical task** of autonomous software systems
- Required for **system identification** & adaptation
- Produce **meaningful data** for free

**In-breadth monitoring**

**In-depth monitoring**

**Decision Making**

- achieves externally specified goals
- measures
- decisions

**Reconfiguration**

- system control inputs
  - queue size, scheduling policy, concurrency level
- actions
- sensors
- effectors

**Target System**

- system outputs
  - response time, utilization, throughput
2010–2014

A focus on power consumption

IN-DEPTH CONTEXT MONITORING
Context

• Energy consumption of ICT (France) [TIC’09]
  – 13.5% in 2009
  – 40% in 2019 (forecast)

• Meanwhile
  – IT devices & datacenters
  – Idle power consumption

• Hardware optimizations
  – Dynamic Voltage/Frequency Scaling (DVFS)
  – Hyper-threading
  – Turboboost
  – Core power states (C-States)

How to identify energy hotspot in applications?

PowerAPI middleware toolkit

Software-defined power meters

- Internal sensors
  - *ProcFS, libpfm4*

- External sensors
  - *PowerSpy, RAPL*

- Formulas
  - *CMOS, DVFS, regression*

- Reporters
  - *GUI, Network, FUSE*

\[
\text{Power}^d_{\text{software}} = \text{CPU}^{\text{utilization}}_{\text{software}} \times \text{Power}^d_{\text{CPU}}
\]
PowerAPI power estimation accuracy

![Graph showing power consumption over time for different applications and tools: x264, freqmine, PowerVM, PowerSpy, and idle power. The graph illustrates the accuracy of PowerAPI in estimating power usage.]
Code-level power monitoring

- **Jalen = PowerAPI++**
  - Exploits process estimations
  - Extracts code-level metrics
  - Monitors utilization per thread (CPU + HDD)
- **2 implementations**
  - Bytecode instrumentation
  - Statistical sampling

\[
\text{Power}_{\text{CPU}}^{\text{thread}} = \frac{\text{Power}_{\text{CPU}}^{\text{software}} \times \text{Utilization}_{\text{CPU}}^{\text{thread}}}{\text{Duration}_{\text{cycle}}}
\]
Jalen instrumentation overhead

Individual Tomcat requests (ms), sampling rate of 10 ms

<table>
<thead>
<tr>
<th></th>
<th>Base</th>
<th>Jalen STS</th>
<th>HPROF</th>
<th>Jalen BCI</th>
<th>JIP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time (ms)</td>
<td>4.157</td>
<td>4.289</td>
<td>4.336</td>
<td>9.532</td>
<td>8.241</td>
</tr>
</tbody>
</table>
Energy dissipation into classes

Jetty web server
> 105,000 LoC

Stress
25 users
100,000 requests
Energy dissipation into methods

- Energy percentage of total
- Number of invocations

![Graph showing energy dissipation into methods](image-url)
In-depth monitoring - key results

• Publications (1 PhD thesis, 3 journals, 2 conferences)

• Collaborations
  – Power consumption estimation in virtualised environments (Univ. of Neuchâtel)
  – Reducing of the power consumption of a HPC cluster (L2EP, Univ. of Lille 1)

• Grants
  – FUI EconHome
  – Inria ADT 2013 (1 engineer)

• Transfer ([http://powerapi.org](http://powerapi.org) - AGPL)
  – Web Energy Archive (funded by ADEME)
  – KaliTerre (dynamic code analysis)
2010–2014

A focus on mobile crowd-sensing

IN-BREADTH CONTEXT MONITORING
Mobile crowd-sensing

• «category of applications that we term mobile crowdsensing, where individuals with sensing and computing devices collectively share data and extract information to measure and map phenomena of common interest» [Ganti’11]

Research challenges

• **Flexibility** [Funf, PRISM]
  - Supporting participative and opportunistic tasks

• **Scalability** [MEDUSA]
  - Supporting both individual and crowd tasks

• **Privacy** [Anonysense]
  - Preserving sensitive data for the participants

• **Energy** [Pogo]
  - Minimizing the power consumption of tasks

• **Incentives** [Anonysense]
  - Engaging participants to contribute along time

PRISM. Das et al. PRISM: platform for remote sensing using smart-phones, *MobiSys’10*
Pogo. Brouwers et al. Pogo, a middleware for mobile phone sensing. *Middleware’12*
Medusa. Ra et al. Medusa: A programming framework for crowd-sensing applications. *MobiSys’12*
APISENSE®: Mobile crowd-sensing

- Crowd-sensing language
  - *JavaScript extension*
  - *Injecting facades*

- Mobile infrastructure
  - *Bee SDK*
  - *Bee App*

- SaaS infrastructure
  - *Hive service*
  - *Honeycomb services*
  - *Virtual sensors* [Chowdhury10]

```javascript
$location.onLocationChange(function(event) {
    $trace.add({
        lat: event.latitude,
        lng: event.longitude,
        signal: $telephony.signalStrength()
    });
});
```

Orchestrating crowd-sensing tasks

**sense**

\[
\text{sense}(\text{function}(\quad) \{ \quad \}) ;
\]

**accept**

\[
\text{accept}(\text{function}(\quad) \{
    \text{if} \ (\$\text{network}.\text{connectionType}() \ == \ ‘\text{Mobile’})
    \ 
    \text{return} \ \{ \ \text{battery} : \ $\text{battery}.\text{level}() \} ;
\}) ;
\]

**recruit**

**ranking**

\[
\text{ranking}(\text{function}(\text{users}) \{ \ 
    \text{return} \ \text{users}.\text{sort}('\text{battery'}); \ 
\}) ;
\]

**coverage**

\[
\text{geoCoverage}(\[
[ [50.614291,3.13282],[50.604159,3.15239] ],
\ ‘500 m’);
\]

\[
\text{timeCoverage}(‘30 \text{ min’},’1 \text{ H’});
\]

\[
\text{duplicate}(10);
\]
Evaluation of APISENSE®

• Objectives
  1. Volume of exchanged data
  2. Energy consumption
  3. Data coverage

• Real mobility traces
  – 10 000 cabs from Beijing

• Comparing 3 strategies
  a. Individual: no coordination (witness)
  b. Coll(500): coverage objective of 500×500 meters
  c. Coll(1000): coverage objective of 1000×1000 meters
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>
APISENSE®: Engaging participants

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

All Campaigns (except Caen)

- 112 371 of data
  - or 39 days of data per user (average)
- 97 volunteers
  - Of which:
    - 64 % male
    - 80 % age under 30
    - 73 % single
    - 35% income < 1500€/m.
- 68 % of students
  - Of which:
    - 43% Computer Sc./ingeneers
    - 22% Communication
- 14 device brands
  - Of which:
    - 41% Samsung
    - 24% LGE
    - 12% Sony
- 48 device models
  - Of which:
    - 12% LGE Nexus 5
- 13 telecom operators
  - Of which:
    - 77% Free, Orange, Bouygues and SFR;
    - 12% unknown
- 14 Android versions
  - Of which:
    - 26% in 4.4.2
Interesting insights for social sciences

1. How much time per day do you use your smartphone?
   - Gap between questionnaire & measurement
   - Underestimate
   - 17 correct estimates
   - Over-estimation

2. How many applications do you use daily?
   - Gap between questionnaire & measurement
   - Underestimate
   - 14 correct estimates
   - Over-estimation

3. How many applications has your smartphone?
   - Gap between questionnaire & measurement
   - Underestimate
   - 6 correct estimates
   - Over-estimation

4. What are the first and second main uses of your smartphone?
   - Score
   - Mail
   - Social Networks
   - Email
   - Video viewing
   - Music
   - Capture Photo / Video
   - Practical Information (Banking, weather, mobility, etc.)
   - 14 correct estimates
   - False estimates
In-breadth monitoring – key results

• Publications (1 PhD thesis, 1 book chapter, 2 conferences)
  
  
  
  – N. Haderer et al. A preliminary investigation of user incentives to leverage crowdsensing activities. In: **PerHot’13.**

• Collaborations
  – **PRACTIC (Inria)**
  – **Priva’Mov (LIRIS / CITI)**
  – **Carbon Footprint Estimation (EQUIPPE, Univ. Lille 1)**
  – **Indoor Air Quality (PC2A, Univ. Lille 1)**
  – **Outdoor Air Quality (LISIC, Univ. Côte d’Opale)**

• Grants
  – **Inria ADT 2012 (1 engineer)**
  – **Inria ADT FOCUS 2014 (3 engineers)**
  – **Microsoft Azure4Research grant**

• Transfer ([http://apisense.com](http://apisense.com))
  – **ip-label (telecom)**
  – **Start-up (ongoing)**
Conclusion

Autonomy of ubiquitous software systems

– *Elastic programming models*
  • Supporting the adaptation of software systems at runtime

– *Feedback control loops*
  • Promoting FCL as first class entities

– *Efficient context monitoring middleware*
  • Enabling in-depth monitoring with PowerAPI
  • Controlling in-breadth monitoring with APISENSE®
Perspective for PowerAPI (1/2)

- PhD thesis of Maxime Colmant (ADEME / Lille 1)
  - Reducing the power consumption of applications running on multi-core architectures
    - Advanced CPU features
    - Virtualisation layers
    - Asymmetric CPUs
    - GPUs
Perspective on APISENSE® (2/2)

• PhD thesis of María Gomez (Inria)
  – *Debugging mobile applications in the wild based on the quality of experience perceived by the crowd*
    • Collecting QoE (traces + feedbacks)
    • Capturing crashes (tests)
    • (Auto) Fixing bugs
    • Deploying patches

![Diagram of Smart App Store process]

1. Ratings & reviews received from the crowd
2. App Monitor receives execution traces and crash reports
3. App Analysis identifies issues
4. Patch Generation creates patches
5. Patch Validation ensures quality
6. Patch Delivery sends patches to users
7. Users rate and review apps
8. Feedback on app performance informs process

• Smart App Store: A system for managing app updates, ensuring user satisfaction.