Verifiable and Resource-Aware Component-based Device Model for IoT

Youakim Badr

INSA-Lyon, LIRIS, France
youakim.badr@insa-lyon.fr
Context: IoT Challenges

CHALLENGES

- Interconnection of physical-world devices

- Strong hardware constraints:
  - CPU frequency ≈ 10 MHz
  - RAM ≈ 10 kB
  - ROM ≈ 100 kB

- Interoperability
  - Bluetooth LE
  - Zigbee
  - IEEE 802.15.4
Motivation case-study

- Home automation for e-Health:
  - **Patient** equipped with **resource constrained** wearable sensors
  - **House** equipped with **environmental sensors** and **actuators**
  - **Computationally unlimited** central **gateway** used for composition
Problematic

How to build smart objects “as a combination” of sensors, controllers, actuators, physical things, taking into account resources

1. **Reusable** component based model:
   - information models, services and a lifecycle
   - cyber-physical properties
   - limited and consumable resources

2. **Resource-aware** verification framework

---

Comprehensive framework to build smart-objects as a combination of connected devices
Reusability in service computing

Services in SOA
- Agnostic
- Loosely coupled
- Remotely executable

Devices in IoT
- Reusable
- Modular
- Scalable

= SOA answers some of the concerns of the IoT

Services in SOA
- Pure programs
- Separation between data and their processing

Devices in IoT
- Hybrid SW/HW
- Self-contained and interoperable

= SOA must be adapted to accurately model IoT Devices
= Logical-based Framework for Devices
Logical-based Framework for Smart Devices

Composable, adaptable and resource-aware connected devices

Challenges
- Liveness of bounded resources
- Artifact Verification (indecidabe) [Bhattacharya et al. BPM 07]
- Structural Contraints, dependancy
- Composition (N-Hard problem) [T. Yu ACM Y Trans on the Web, 2007]
- Dynamic Environment
- Incomplete Information

IoT

Specification
- Resources
- Contraintes

Composition
- Service Composition
- Composition of connected devices

Self-Adaptable
- Small scale
- Large scale

Validation/Verification

Automated and correct

Safety
Component-based Connected Device Model

Ambient healthcare component

Closed feedback loop composition logic
A connected device \( \mathbf{D}_i = < id, R, Q, QoS, S > \)

- **id** is the identifier of the connected device \( \mathbf{D}_i \).
- **R** set of consumable resources: \( R = R_v \cup R_r = \{ r^n \} \cup \{ r^{[n]} \} \)
  - \( R_v \) is the set of vanishing resources
    - \( n=1, r^n \) is consumed one time
    - \( n>1, r^n \) is consumed more than one time (\( n \) is bounded)
    - \( n=! , r^n \) is consumed unlimited number of times (\( n \) is unbounded)
  - \( R_r \) is the set of renewal resources
    - \( n=1, r^{[n]} \) is consumed at most \( n \) times (can be regenerated when disposed)

- \( S = \{ s_1, s_2, ... \} \) is the set of services.
  - A service \( s_i = < I \cup O, R, Q, QoS_i, E > \)
    - \( I = \{ i_1, i_2 ... \} \), is a set of input messages \( (i_i) \).
    - \( O = \{ o_1, o_2 ... \} \), is a set of output messages \( (o_i) \).
    - \( E \) = is an exception

- \( Q = \) set of states = \( \{ q_1, q_2, q_3, ... \} \)
- \( QoS = < Q_1, Q_2, ..., Q_n > \) is the set of qualitative and quantitative QoS attributes \( (q_i) \).

\[
\text{id } \otimes \text{ R}_v \otimes \text{ R}_r \otimes \text{ Q } \otimes \text{ I } \rightarrow ((\text{O } \otimes \text{ R}_r) \oplus \text{ E}) \otimes \text{ Q}
\]
ECG sensor (E) :  HR() \rightarrow hr \\
HRV(hr300) \rightarrow hrv = \{SSDN, RMSSD, LF-HF, NormLF\} \\
buffer(hr, timestep) \rightarrow hr300 // timestep global variable (context)

Actimetry sensor (A) :  Activity() \rightarrow activity = \{running, walking, sleeping, sitting\}

Electrodermal sensor (D) :  Dermal() \rightarrow srr // skin resistance response

MailServer (M) :  send(addresses) \rightarrow email

Phone (P) :  send(riskLevel, phoneNumbers) \rightarrow sms \\
call(RiskLevel, phoneNumber) \rightarrow voiceMsg

Inertial measurement unit (IMU):  position(on) \rightarrow (x, y, z)

Occupancy sensor (O_1, ... O_n) :  presence(on) \rightarrow (Ispresent, room) \\
PositionRoom(room) \rightarrow (x, y, z)

Controller@Home (CH):  Predit(hrv, activity, ?srr) \rightarrow riskLevel \\
Control(riskLevel, (x, y, z)) \rightarrow Phone.call xor (?MailServer.send | | Phone.Alarm

Controller@Phone (CP):  Predit(hrv, activity, srr) \rightarrow riskLevel \\
Control service: control(riskLevel, (x, y, z)) \rightarrow [send | | call | | alarm] xor Mail.Send

Vanishing resources:  \text{R}_v = \{\text{battery (200mAh), security certificate (expiration date), password (80 times), ...}\} 

Renewal resources:  \text{R}_r = \{\text{memory (64Kb), bandwidth (100Mbs), ...}\} 

QoS = \{\text{response\_time, cost(euros), concurrentRequests(capacity), throughput(latency/capacity), dataEncryption (y/n), Connectivity([BL | Wifi | wired]), Reliability, ...}\}
BEGIN; andSplit(Actimetry.Activity(activity), Electrodermal.Dermal(srr), ECG.HR(hr));
seq[ECG.HR.hr = ECG.buffer.hr];
ECG.buffer(hr, timestep)(hr300);
seq[ECG.buffer.hr = ECG.HRV.hr300];
ECG.HRV(hr300);
xor[switch(context)(home ; outside)] (?
\begin{itemize}
    \item ?home(?
        Controller@Home.Predict(hrv, activity, ?srr)(riskLevel);
        seq[Controller@Home.Predict.riskLevel= Controller@Home.Control.riskLevel]);
    \item Controller@Home.Control(riskLevel, (x, y, z));
        xorSplit[switch(SGM)](OK ; KO)(?
            Controller@Home.Control.riskLevel);
    \item Phone.Alarm(phoneNumber)(voiceMsg)
    \end{itemize}
\end{itemize}
)
\begin{itemize}
    \item ?KO(?
        andSplit(?
            Phone.Alarm(phoneNumber)(voiceMsg),
            MailServer.Send(addresses)(email)
        )
        xorJoin(email, voiceMsg)
    \end{itemize}
\end{itemize}
\begin{itemize}
    \item ?outside(?
        Phone.Call(RiskLevel, phoneNumbers)(voiceMsg)
    )
\end{itemize}
}
\begin{itemize}
    \item xorJoin(voiceMsg or email, voiceMsg)
\end{itemize}

Composition Patterns
- Sequence Pattern: $s_i ; s_j$
- Parallel Pattern: $\text{andSplit}(s_i , s_j , s_k )$
- Selection Pattern: $\text{xorSplit[condition]}(s_i , s_j , s_k )$
- Iteration Pattern: $\text{iter[condition]}(s_i)$
QoS Composite Connected Devices (II)

<table>
<thead>
<tr>
<th>QoS Attribute Name</th>
<th>Measure Scales</th>
<th>Composition Pattern Calculation Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectivity(CO)</td>
<td>Nominal</td>
<td>Sequence: $\bigcap_{i=1}^{n} co_i$</td>
</tr>
<tr>
<td>Security (SE)</td>
<td>Ordinal</td>
<td>min(se$_i$)</td>
</tr>
<tr>
<td>Rating Point (RP)</td>
<td>Interval</td>
<td>min(rp$_i$)</td>
</tr>
<tr>
<td>Response Time (RT)</td>
<td>Ratio</td>
<td>$\sum_{i=1}^{n} rt_i$</td>
</tr>
<tr>
<td>Reliability (RE)</td>
<td>Ratio</td>
<td>$\prod_{i=1}^{n} re_i$</td>
</tr>
<tr>
<td>Price (P)</td>
<td>Ratio</td>
<td>$\sum_{i=1}^{n} p_i$</td>
</tr>
</tbody>
</table>
Composition Constraints (III)

- **Structural constraints**
  - ECG.buffer; ECG.HRV

- **Constraint constraints**
  - **Local Constraints**
    - \( c_r = < \text{Phone.battery}, \exists, \text{Low} > \)
  - **Global Constraints**
    - \( c_{rg} = < \text{HeartAttac.response_time}, <, 2 \text{ min}> \)

- **Dependency constraints**
  - Preferred, Excluded, ...
Resilient connected object model (IV)

- Reactive vs proactive strategies
- Moving Target Technique
- Generation of Variants
  - Substitution
  - Replication
  - Contextual Variations du contexte
  - Topologies
    - Composition
    - Decomposition
    - ...

Logical-based Framework for Smart Devices

Composable, adaptable and resource-aware connected devices

**IoT**

**Specification**
- Resources
- Contraintes

**Composition**
- Service Composition
- Composition of connected devices

**Self-Adaptable**
- Small scale
- Large scale

**Validation/Verification**
- Liveness of bounded ressources
- Artifact Verification (indecidable)  
  [Bhattacharya et al. BPM 07]

**Automated and correct**
- Structural Contraints, dependancy
- Composition (N-Hard problem)  
  [T. Yu ACM Y Trans on the Web, 2007]

**Safety**
- Dynamic Environment
- Incomplete Information

**Challenges**

**Logical Frameworks**: Abella, Beluga, Hybrid, Twelf, ...

**Formal reasoning systems**: Coq, Isabelle/HOL

**Linear Logic** [Girard 1986]: linprover, l2coq, Ilprover, TATU and Quati,

- Resource-aware lambda-calculus  
  [Lafont 1993]
- Proofs-as-processes in RAPS (ILL)  
  [Rao et al. 2006]
- Proofs-as-processes in HOL Light (CLL)  
  [Papapanagiotou et al. 2011]
- Constraint programming  
  [Rao et al. 2006]
Logic-based Framework for Smart Devices
Composable, adaptable and resource-aware connected devices

• **Specification of bounded resources**
  • spatial and temporal resources, energy consumption, storage and computational ...

• **Constraints Specifications to all possible compositions**
  • inter-objects structural constraints, dependency, exclusion, ...

• **Automated theorem proving**
  • automated and correct Composition
    • Find a proof, satisfying constraints
    • Control search strategy
  • Adaptation and safety
    • Find all “possible” proofs (if possible)

• **Proofs Verification**
  • Proof-as-program (Curry-Howard)
    • Export certified proofs

• **Formal Computational Specifications with Subexponential Logic**
  - Specify **Subexponential logic framework**
  - Automate proof search (certify and export proof)
Verifiable composition – Taste of Linear Logic

Linear Logic (J.Y. Girard 1986)
- A ::= P | A ⊸ A | A ⊗ A | A ⊕ A | A & A | !A | 1

Antony Hoare (1985)'s Vending Machine example
- Linear Logic: $1 ⇒ candy
- Classical or Intuitionistic logic
  - (A and A ⇒ B) ⇒ A ∧ B
  - ($1 and $1 ⇒ candy) ⇒ $1 ∧ candy

A Logic of Resources
- A ⊸ B = transform resource A into resource B
  - $1 ⇒ candy
- A ⊗ B = Multiplicative conjunction consumes simultaneous resources
  - $3 ⇒ (candy ⊗ chips ⊗ drink) // $3 := $1 ⊗ $1 ⊗ $1
  - $1 ⇒ (candy ⊗ chips ⊗ drink) // wrong

- A & B = Additive conjunction represents alternative occurrence of resources
  - $1 ⇒ (candy & chips & drink)
  - $3 ⇒ (candy & chips & drink)
- A ⊕ B = Additive disjunction represents alternative occurrence of resources
  - $1 ⇒ (candy ⊕ chips ⊕ drink)
Composite devices (wf) in GSM model

• **Lifecycle:**
  - **Guards:** entry point of a stage.
  - **Stage body:** contains one or multiple services.
  - **Milestones:** events that deactivates the stage body.

Ambient healthcare component lifecycle derived from the composition of multiple sensors
Encoding GSM with linear logic

acquiring: \( CPAK \leftarrow RRR \otimes HRR \otimes AR \otimes SRR \)
checkingSR: \( CPU \otimes SRR \leftarrow SROK \)
chekingRR: \( CPU \otimes RRR \leftarrow RROK \)
chekingAcc: \( CPU \otimes AR \leftarrow AccOK \)
chekingHR: \( CPU \otimes HRR \leftarrow HROK \)
computing: \( RROK \otimes AccOK \otimes SROK \otimes HROK \leftarrow BE \)
Verifiable composition

Linear logic statements
- acquiring: $CPAK \vdash RRR \otimes HRR \otimes AR \otimes SRR$
- checkingSR: $CPU \otimes SRR \vdash SROK$
- chekingRR: $CPU \otimes RRR \vdash RROK$
- chekingAcc: $CPU \otimes AR \vdash AccOK$
- chekingHR: $CPU \otimes HRR \vdash HROK$
- computing: $RROK \otimes AccOK \otimes SROK \otimes HROK \vdash BE$

Linear logic inference rules
- $A \vdash A$ (axiom)
- $\vdash 1$ (1R)
- $\Gamma \vdash A \Delta \vdash B \quad (\otimes R)$
- $\Gamma, A \vdash C \quad (\otimes L)$
- $\Gamma, A \vdash B \quad (-\circ R)$
- $\Gamma, A \Delta \vdash C \quad (\circ L)$

Linear logic theorem
- $CPU^4 \otimes CPAK \vdash BE$

Proof tree = composition verified
LL Proof

• The theorem to be proven is: \( CPU^4 \otimes CPAK \vdash BE \)

• The proof tree of the ambient healthcare composite component is:
Proof tree interpretation

• Proof tree can further be interpreted in terms of composition:
  • The \textit{cut} rule corresponds to the serial composition of stages.
  • The $\otimes R$ and $\otimes L$ rules correspond to the parallel composition of stages.
Q /A