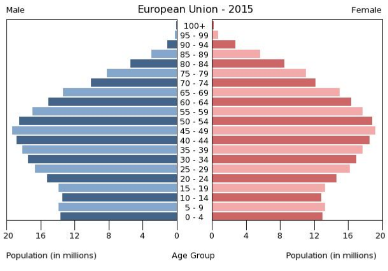
Mechanism for Optimizing the Reasoning for Activity Recognition: Application for Ambient Assisted Living for Elderly People

Romain Endelin

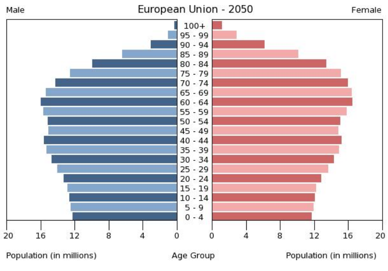
Laboratoire d'Informatique, Robotique et Microélectronique de Montpellier (LIRMM), FRANCE
Institut Mines-Télécom, FRANCE

June 2^{nd}, 2016

An Ageing Population



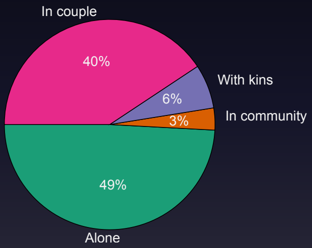
Population pyramids in Europe — 2015



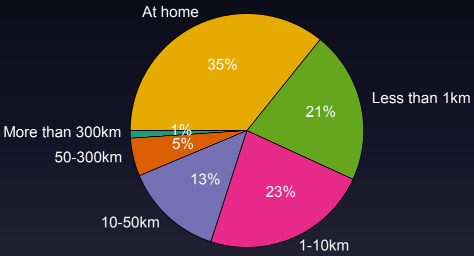
Population pyramids in Europe — 2050

An Ageing Population

Study performed in 2014, with 120 elderly people and their caregivers[1].



Person living with elderly people

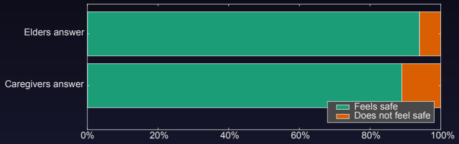


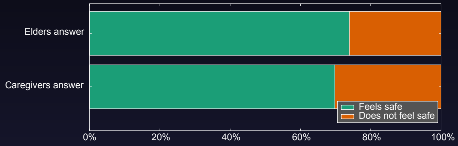
Distance separating elderly people from their caregiver

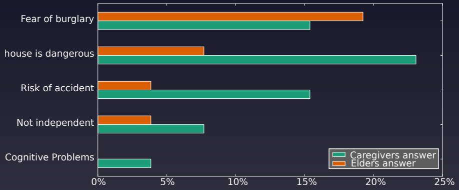
1 Mounir Mokhtari, Romain Endelin, Hamdi Aloulou, and Thibaut Tiberghien. "Measuring the impact of ICTs on the quality of life of ageing people with mild dementia". In: ICOST 2014. Springer, 2014, pp. 103–109.

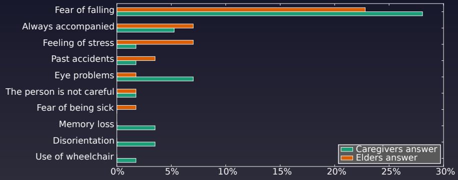
An Ageing Population

Study performed in 2014, with 120 elderly people and their caregivers[1].







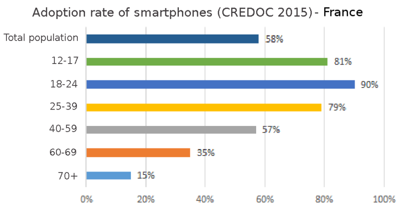


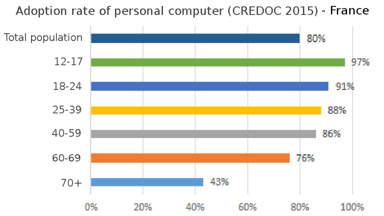
Feeling of security inside elderly people's house

Feeling of security outside elderly people's house

1 Mounir Mokhtari, Romain Endelin, Hamdi Aloulou, and Thibaut Tiberghien. "Measuring the impact of ICTs on the quality of life of ageing people with mild dementia". In: ICOST 2014. Springer, 2014, pp. 103–109.

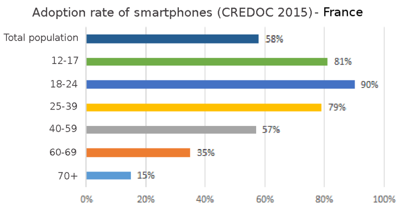
Adoption of New Technologies

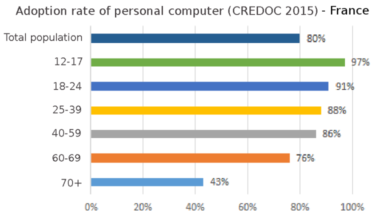




- Causes of non-adoption from elderly people:
  - Learning curve;
  - Lack of habits;
  - Do not see the interest;
  - Systems are not designed with elderly people in mind.

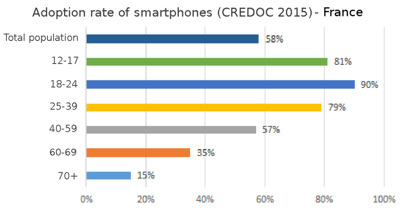
Adoption of New Technologies

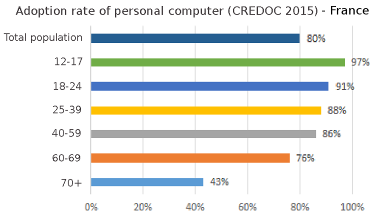




- Causes of non-adoption from elderly people:
  - Learning curve;
  - Lack of habits;
  - Do not see the interest;
  - Systems are not designed with elderly people in mind.

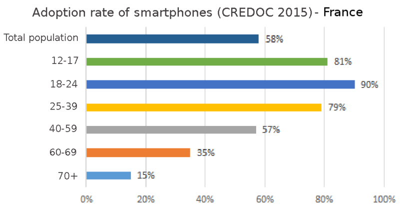
Adoption of New Technologies

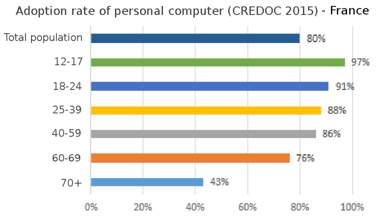




- Causes of non-adoption from elderly people:
  - Learning curve;
  - Lack of habits;
  - Do not see the interest;
  - Systems are not designed with elderly people in mind.

Adoption of New Technologies





- Causes of non-adoption from elderly people:
  - Learning curve;
  - Lack of habits;
  - Do not see the interest;
  - Systems are not designed with elderly people in mind.

Pervasive Computing

Definition

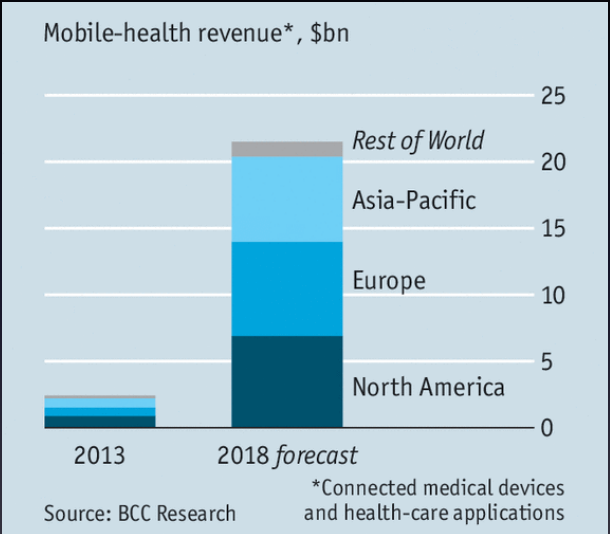
"The age of calm technology, when technology recedes into the background of our lives"

Mark Weiser, 1991

The Benefits of Pervasive Computing:

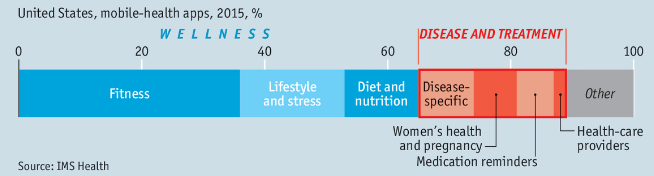
- Reduced learning curve;
- Adapted for people with disabilities;
- Customized interactions for each users.

Opportunity — eHealth



The growth of Mobile Health

Opportunity — eHealth



Mobile Health — From Well-Being to Disease Treatments

Ambient Assisted Living

- Large number of existing implementations;
  - Only a handful of them have real world usage[2].

Recommended focus areas:[2]

- Design methodologies;
- Security;
- User experience;
- Usability;
- Accuracy.

Existing solutions:

- UniversAAL;
- OpenAAL;
- OpenHAB;
- CASAS;
- DOMUS;
- …

2 Mukhtiar Memon, Stefan Rahr Wagner, Christian Fischer Pedersen et al. "Ambient assisted living healthcare frameworks, platforms, standards, and quality attributes". In: Sensors 14.3 (2014), pp. 4312–4341

Context

The team — PAWM International

- Top-down approach:
  - Focus on practical applications;
  - *Pull* challenges from the use-case.

Initial situation

- UbiSMART V2, a modular framework for Ambient Assisted Living[3];
- A Semantic-based Reasoning for Activity Recognition[4].

3 Hamdi Aloulou. “Framework for ambient assistive living: handling dynamism and uncertainty in real time semantic services provisioning”. PhD thesis. Evry, Institut National des Télécommunications, 2014

4 Thibaut Tiberghien. “Strategies for context reasoning in assistive livings for the elderly”. PhD thesis. Evry, Institut National des Télécommunications, 2013

Outline

Use-Case: Moving Towards a Real and Scalable Deployment

Saint-Vincent-de-Paul, Nursing Home, ***Occagnes, FRANCE***

The Deployment

- **Starting:** September 2013
    - Active until today
- Nursing home
    - 5 rooms deployed
- Individual houses
    - Up to 3 individual deployed

Chaire Quality of Life. Mutuelle Générale; AG2R La Mondiale.
VHP Interactive.

Scientific Challenges

- A framework for Ambient-Assisted Living:
  - From *proof-of-concept* to *production-ready* implementation;
- Qualify and improve Activity Recognition.

Outline

UbiSMART — Challenges

- ➡ A framework for Ambient-Assisted Living:
  - Simplify the installation process;
  - Seamlessly extend to multiple deployments;
  - Design for the end-users;
- Qualify and improve Activity Recognition.

UbiSMART — Challenges

- ➡ A framework for Ambient-Assisted Living:
  - Simplify the installation process:
    - Install a gateway within the house;
    - Define a right set of sensors;
    - Communicate with the server;
    - Maintain the system.
  - Seamlessly extend to multiple deployments;
  - Design for the end-users;
- Qualify and improve Activity Recognition.

UbiSMART — Challenges

- ➡ A framework for Ambient-Assisted Living:
  - Simplify the installation process;
  - Seamlessly extend to multiple deployments:
    - Automatic configuration of the gateway;
    - Monitoring;
    - Improve performances;
  - Design for the end-users;
- Qualify and improve Activity Recognition.

UbiSMART — Challenges

- ➡ A framework for Ambient-Assisted Living:
  - Simplify the installation process;
  - Seamlessly extend to multiple deployments;
  - Design for the end-users:
    - User-Friendly services.
    - Secured access for the users;
- Qualify and improve Activity Recognition.

UbiSMART — Challenges

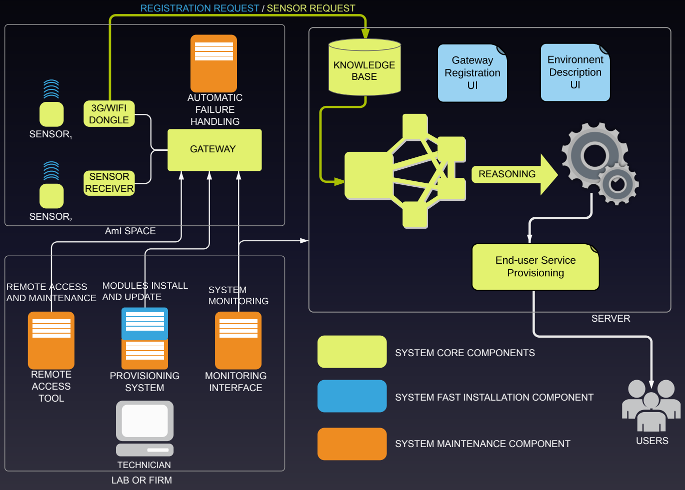
- ➡ A framework for Ambient-Assisted Living:
  - Simplify the installation process:
    - Install a gateway within the house;
    - Define a right set of sensors;
    - Communicate with the server;
    - Maintain the system.
  - Seamlessly extend to multiple deployments:
    - Automatic configuration of the gateway;
    - Monitoring;
    - Improve performances;
  - Design for the end-users:
    - User-Friendly services.
    - Secured access for the users;
- Qualify and improve Activity Recognition.

Personal contribution of this thesis
In close collaboration with other researchers of the team
Work mostly carried by other researchers of the team

UbiSMART — Challenges

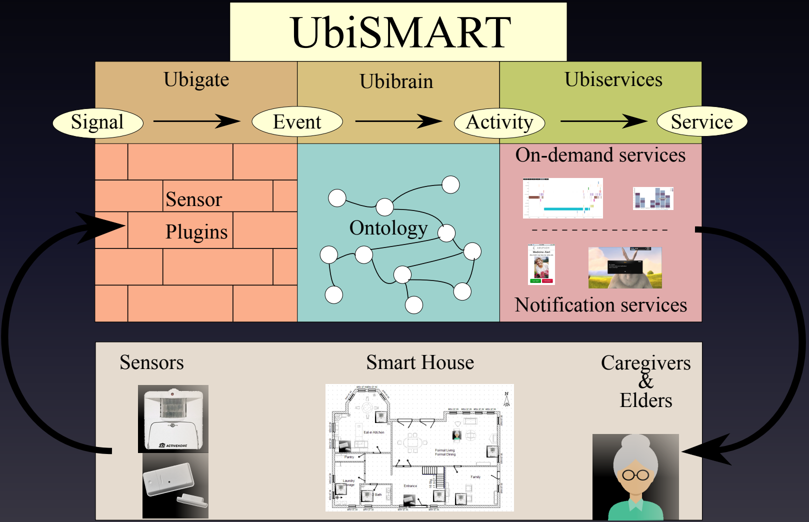
- ➡ A framework for Ambient-Assisted Living:
  - Simplify the installation process:
    - Install a gateway within the house; $\Longleftarrow$
    - Define a right set of sensors; $\Longleftarrow$
    - Communicate with the server; $\Longleftarrow$
    - Maintain the system.
  - Seamlessly extend to multiple deployments:
    - Automatic configuration of the gateway; $\Longleftarrow$
    - Monitoring;
    - Improve performances;
  - Design for the end-users:
    - User-Friendly services. $\Longleftarrow$
    - Secured access for the users;
- Qualify and improve Activity Recognition.

Response to the challenges



Deployment in a house

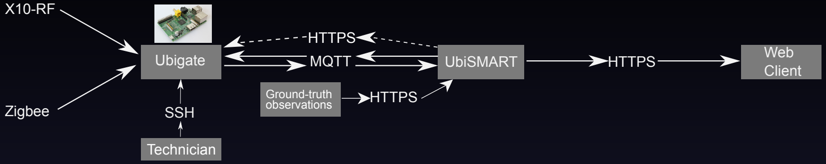
The New UbiSMART Framework



Improvements over the previous UbiSMART

	UbiSMART V2	**UbiSMART V3**
Language	Java	Javascript
Framework	OSGi	NodeJS/Express
Environment	Command-Line	Web-Application
Paradigm	Service-Oriented Architecture	Model/View/Controller
Modularity	Service-based	Plugin-based
Reasoner	✓	✓
Sensor events	✓	✓
Service delivery	~	✓
User interface	✗	✓
Encryption	✗	✓
Multiple houses	✗	✓
User privileges	✗	✓
Data persistence	✗	✓
Semantic Plug&Play	✗	✓

Communication Protocols



Roles:

- MQTT:
  - **Sensor events;**
  - Sensor discovery;
  - Service delivery;
  - Monitoring;
  - Configuration.
- HTTPS:
  - Gateway discovery;
  - Ground-truth observations.
- SSH:
  - Provisioning;
  - Maintenance.

MQTT messages:

- Topic:

```
house/1/marmitek/sensor/A1
```

- Payload:

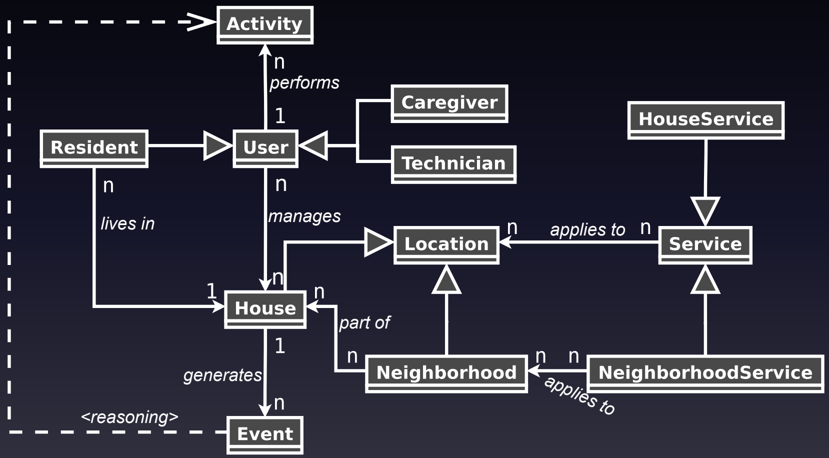

```
{
  "date": "2016-05-17T09:06:00.116+02:00",
  "house": 1,
  "sensor": "A1",
  "value": "off"
}
```

Data Input

Input — Sensor events

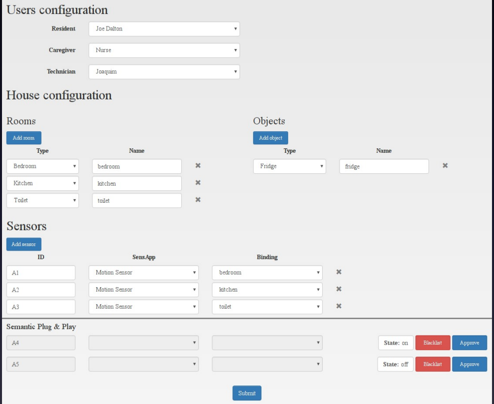
date,	house,	sensor,	value
2014-09-23 13:33:00.000000,	1,	A1,	on
2014-09-23 13:33:00.000000,	2,	A1,	on
2014-09-23 13:37:00.000000,	3,	A1,	on
2014-09-23 13:44:40.000000,	5,	C2,	On
2014-09-23 14:05:36.000000,	3,	a819880,	normal
2014-09-23 14:05:42.000000,	5,	C2,	On
2014-09-23 14:05:49.000000,	5,	C2,	On
2014-09-23 14:05:53.000000,	3,	a819880,	alert
2014-09-23 14:05:55.000000,	3,	a819880,	normal
2014-09-23 14:05:57.000000,	3,	a819880,	alert
2014-09-23 14:05:59.000000,	3,	a819880,	normal
2014-09-23 14:06:07.000000,	3,	a819880,	alert
2014-09-23 14:06:10.000000,	3,	a819880,	normal
2014-09-23 14:06:15.000000,	3,	a819880,	alert
2014-09-23 14:06:32.000000,	5,	C1,	On
2014-09-23 14:06:36.000000,	3,	a819880,	normal
2014-09-23 14:06:37.000000,	3,	a819880,	alert
2014-09-23 14:06:38.000000,	3,	a819880,	alert
2014-09-23 14:07:00.000000,	3,	A1,	On
2014-09-23 14:07:06.000000,	3,	A1,	On

UbiSMART Model



Most of the credit for development go to Dr. Thibaut Tiberghien and Mr Joaquim Bellmunt Montoya.

Semantic Plug&Play

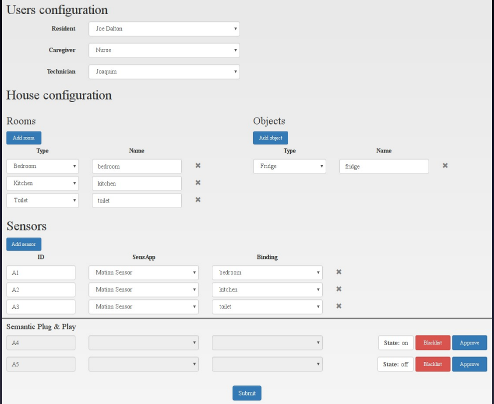


House declaration — Homedesc

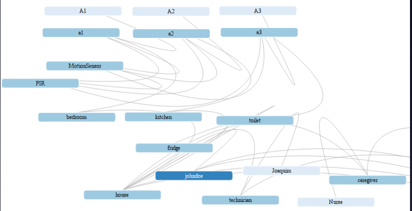
```
hom:johndoe rdf:type qol:Resident.
hom:johndoe qol:residentOf hom:house.
hom:johndoe qol:id "4"^^xsd:integer.
hom:johndoe qol:name "Joe Dalton"^^xsd:string.
hom:caregiver rdf:type qol:Caregiver.
hom:caregiver qol:caregiverOf hom:house.
hom:caregiver qol:id "5"^^xsd:integer.
hom:caregiver qol:name "Nurse"^^xsd:string.
hom:technician rdf:type qol:Technician.
hom:technician qol:technicianOf hom:house.
hom:technician qol:id "3"^^xsd:integer.
hom:technician qol:name "Joaquim"^^xsd:string.
hom:bedroom rdf:type qol:Bedroom.
hom:bedroom qol:partOf hom:house.
hom:kitchen rdf:type qol:Kitchen.
hom:kitchen qol:partOf hom:house.
hom:toilet rdf:type qol:Toilet.
hom:toilet qol:partOf hom:house.
hom:fridge rdf:type qol:Fridge.
hom:fridge qol:locatedIn hom:house.
hom:a1 rdf:type hom:MotionSensor.
hom:a1 qol:id "A1"^^xsd:string.
hom:a1 qol:deployedIn hom:bedroom.
hom:a2 rdf:type hom:MotionSensor.
hom:a2 qol:id "A2"^^xsd:string.
hom:a2 qol:deployedIn hom:kitchen.
hom:a3 rdf:type hom:MotionSensor.
hom:a3 qol:id "A3"^^xsd:string.
hom:a3 qol:deployedIn hom:toilet.
```

Ontology output

Semantic Plug&Play

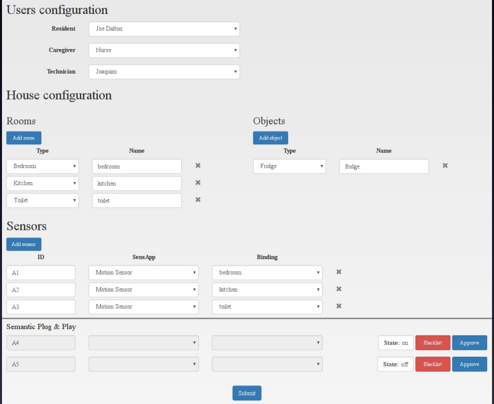


House declaration — Homedesc



Ontology visualization

Semantic Plug&Play



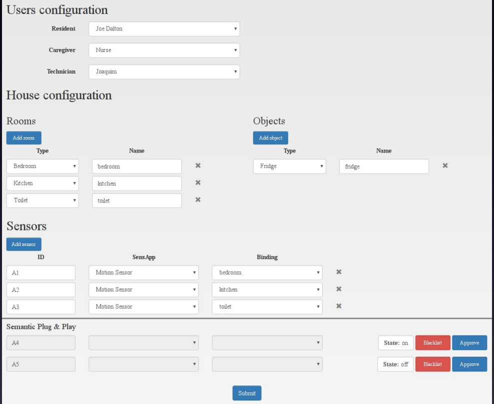
House declaration — Homedesc


```
{
 "houses":[
  {
   "id":1,
   "sensors":[
    {
     "id":"A1",
     "binding":"bedroom",
     "bindingType":"Room"
    },
    {
     "id":"A2",
     "binding":"kitchen",
     "bindingType":"Room"
    },
    {
     "id":"A3",
     "binding":"toilet",
     "bindingType":"Room"
    }
   ]
  }
 ],
 "blacklist":[

 ]
}
```


Gateway output

Semantic Plug&Play



House declaration — Homedesc

Benefits

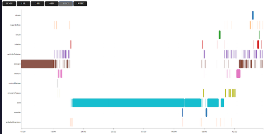
- The deployment can be performed by a non-technician;
- Ontologies are specific per-house.

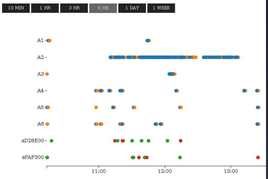
Semantic Plug&Play

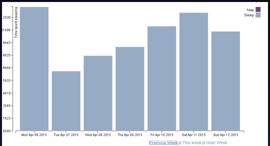
Demo…

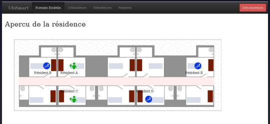
UbiSMART Service Provisioning

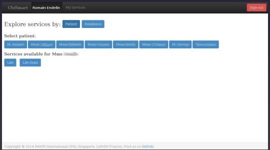
Use case — Visualizing activities

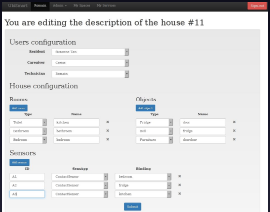


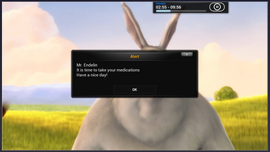












Validation — Installation

	Without our approach	**With our approach**
Background Work	1h	1h
Active Work	45min	5min
Installer	Expert	Non-Expert
Deployment Time	4h	20min
Installer	Expert	Non-Expert
Upgrade process	5-20min per gateway	0

Validation — Maintenance

- Improvement on the maintenance process
  - Daily Maintenance Duration = $N \times (D + R \times (1 - A) \times T)$
  - Expected Downtime = $C + (1 - A) \times T$

	Without our approach	**With our approach**
Daily Maintenance Duration	2h 5min	15min
Downtime	12 hours	6 minutes

Validation — Maintenance

- Improvement on the maintenance process
  - Daily Maintenance Duration = $N \times (D + R \times (1 - A) \times T)$
  - Expected Downtime = $C + (1 - A) \times T$
- Where:
  - A: The probability that a house can be fixed automatically after a failure;
  - T: The time required to manually fix a house;
  - N: The number of deployed houses;
  - D: The expected duration until a faulty house is noticed.
  - R: The daily probability of having a failure in a house;
  - C: The time required to check one house;

	Without our approach	**With our approach**
A	0	0.8
T	30min	
N	50 houses	
D	12h	10 sec
R	0.05	
C	1min	0min
Daily Maintenance Duration	2h 5min	15min
Downtime	12 hours	6 minutes

Validation — Maintenance

- Improvement on the maintenance process
  - Daily Maintenance Duration = $N \times (D + R \times (1 - A) \times T)$
  - Expected Downtime = $C + (1 - A) \times T$
- Where:
  - A: The probability that a house can be fixed automatically after a failure;
  - T: The time required to manually fix a house;
  - N: The number of deployed houses;
  - D: The expected duration until a faulty house is noticed.
  - R: The daily probability of having a failure in a house;
  - C: The time required to check one house;

	Without our approach	**With our approach**
A	0	0.8
T	30min	
N	50 houses	
D	12h	10 sec
R	0.05	
C	1min	0min
Daily Maintenance Duration	2h 5min	15min
Downtime	12 hours	6 minutes

Outcome: A Framework for Ambient Assisted Living[5,6]

Person	Start of deployment	End of deployment	Events	Activities inferred
Resident 1	23 Sep 2014	14 Mar 2015	29,194	6,491
Resident 2	15 Mar 2015	26 Aug 2015	34,680	10,101
Resident 3	28 Aug 2015	Ongoing	5,210	1,243
Resident 4	23 Sep 2014	06 Aug 2015	63,715	12,897
Resident 5	07 Aug 2015	Ongoing	2,891	428
Resident 6	23 Sep 2014	Ongoing	85,231	15,946
Resident 7	23 Sep 2014	Ongoing	44,102	10,332
Resident 8	25 Sep 2014	03 Jun 2015	10,576	2,673
Resident 9	04 Jun 2015	Ongoing	23,536	1,636
House 1	24 Sep 2014	02 Jan 2016	160,275	22,604
			299,135	84,351

Amount of data obtained in our deployment

5 Joaquim Bellmunt, Thibaut Tiberghien, Mounir Mokhtari, Hamdi Aloulou, and **Romain Endelin**. "Technical Challenges Towards an AAL Large Scale Deployment". In: ICOST 2015. Springer International Publishing, 2015, pp. 3–14.

6 Hamdi Aloulou, Mounir Mokhtari, Bessam Abdulrazak, **Romain Endelin**, Thibaut Tiberghien, and Joaquim Bellmunt. "Simplifying Installation and Maintenance of Ambient Intelligent Solutions Toward Large Scale Deployment". In: ICOST 2016. Springer, 2016.

Outcome: A Framework for Ambient Assisted Living[5,6]

Person	Start of deployment	End of deployment	Events	Activities inferred
Resident 1	23 Sep 2014	14 Mar 2015	29,194	6,491
Resident 2	15 Mar 2015	26 Aug 2015	34,680	10,101
Resident 3	28 Aug 2015	Ongoing	5,210	1,243
Resident 4	23 Sep 2014	06 Aug 2015	63,715	12,897
Resident 5	07 Aug 2015	Ongoing	2,891	428
Resident 6	23 Sep 2014	Ongoing	85,231	15,946
Resident 7	23 Sep 2014	Ongoing	44,102	10,332
Resident 8	25 Sep 2014	03 Jun 2015	10,576	2,673
Resident 9	04 Jun 2015	Ongoing	23,536	1,636
House 1	24 Sep 2014	02 Jan 2016	160,275	22,604
			299,135	84,351

Amount of data obtained in our deployment

5 Joaquim Bellmunt, Thibaut Tiberghien, Mounir Mokhtari, Hamdi Aloulou, and **Romain Endelin**. "Technical Challenges Towards an AAL Large Scale Deployment". In: ICOST 2015. Springer International Publishing, 2015, pp. 3–14.

6 Hamdi Aloulou, Mounir Mokhtari, Bessam Abdulrazak, **Romain Endelin**, Thibaut Tiberghien, and Joaquim Bellmunt. "Simplifying Installation and Maintenance of Ambient Intelligent Solutions Toward Large Scale Deployment". In: ICOST 2016. Springer, 2016.

Outline

Activity Recognition — Challenges

- ✓ A framework for Ambient-Assisted Living;
- ➡ Qualify and improve Activity Recognition:
  - Perform Activity Recognition;
  - Qualify the reasoning;
  - Make the reasoning more accurate;
  - Facilitate prototyping;
  - Include a learning process.

Activity Recognition — Challenges

- ✓ A framework for Ambient-Assisted Living;
- ➡ Qualify and improve Activity Recognition:
  - Perform Activity Recognition;
  - Qualify the reasoning;
  - Make the reasoning more accurate;
  - Facilitate prototyping;
  - Include a learning process.

Existing before the beginning of this thesis
Personal contribution of this thesis
Beyond this thesis' scope

Activity Recognition — Introduction

- **Purpose:** Accurately recognize the user's activity, in order for the system to provide an adequate reaction.
- **Input:** A sequence of sensor events.
- **Method:** Divided between two approaches:

Data-driven

- Powerful computations, using machine-learning methods.

Knowledge-driven

- Ontological representation of knowledge.

Activity Recognition — Introduction

- **Purpose:** Accurately recognize the user's activity, in order for the system to provide an adequate reaction.
- **Input:** A sequence of sensor events.
- **Method:** Divided between two approaches:

Data-driven

- Powerful computations, using machine-learning methods.

Knowledge-driven

- Ontological representation of knowledge.

Activity Recognition — Introduction

- **Purpose:** Accurately recognize the user's activity, in order for the system to provide an adequate reaction.
- **Input:** A sequence of sensor events.
- **Method:** Divided between two approaches:

Data-driven

- Powerful computations, using machine-learning methods.

Knowledge-driven

- Ontological representation of knowledge.

Choosing knowledge-driven methods:

- Does not require datasets;
- Specific to each user;
- Adapted for complex-systems.

UbiSMART's Original Reasoning Engine

- Using ontological models:
  - Semantic web technologies;
  - N3 formalism;
  - Euler Rule Engine.
- Includes both *model* and *reasoning*:

Model

```
hom:resident_a a :Resident ;
    :detectedIn hom:outside ;
    :name "A" ;
    :residentIn hom:house_a .
hom:sensor_bedroomMotion a :MotionSensor ;
    :deployedIn hom:room_bedroom ;
    :id "Bedroom Motion"@en .
hom:sensor_mainDoorSensor a :ContactSensor ;
    :attachedTo hom:door_bedroom-outside ;
    :id "Main Door Sensor"@en .
```

Reasoning

```
{
  ?u qol:detectedIn ?r.
  ?r a qol:Livingroom.
  ?r qol:motionMeasured ?m.
  ?m math:notLessThan 2
} => {
  hom:watchtv :getScore 7.
}.
```

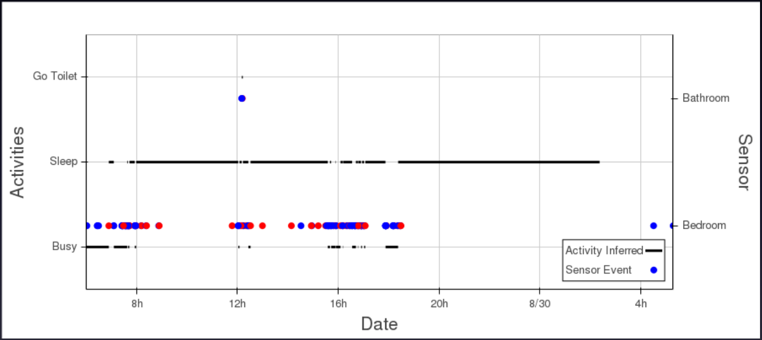
Problems Observed

- Faulty inference due to a **poorly defined rule**;
- Reasoner trying to be **too fine-grained**;
- Reasoner being **too coarse-grained**.

Outline

Problems Observed — 1/2

- ➡ Faulty inference due to a poorly defined rule:



- Reasoner trying to be too fine-grained;
- Reasoner being too coarse-grained.

Making Rules Less Error-Prone

Rules Factory

- Going from this…

```
{?u qol:detectedIn ?r. ?r a qol:Outside. hom:clock qol:hasValue ?t.
  ?t func:hours-from-dateTime ?h.
  ?h math:notLessThan 22. ?h math:lessThan 7
} => {hom:runAway :getScore 9}.
{?u qol:detectedIn ?o. ?o a qol:Outside. ?u qol:inRoomFor ?d.
?d math:notLessThan 18000
} => {hom:runAway :getScore 9}.
```

Making Rules Less Error-Prone

Rules Factory

- Going from this…

```
{?u qol:detectedIn ?r. ?r a qol:Outside. hom:clock qol:hasValue ?t.
 ?t func:hours-from-dateTime ?h.
 ?h math:notLessThan 22. ?h math:lessThan 7
} => {hom:runAway :getScore 9}.
{?u qol:detectedIn ?o. ?o a qol:Outside. ?u qol:inRoomFor ?d.
?d math:notLessThan 18000
} => {hom:runAway :getScore 9}.
```

- …to that:

```
[{
   "name": "Run away",
   "conditions": {
      "hour_of_the_day": [22, 7],
      "detected_in": {"room": "room:Outside"},
      "min_duration": "5min"
   }
}]
```

Making Rules Less Error-Prone

Rules Factory

- Going from this…

```
{?u qol:detectedIn ?r. ?r a qol:Outside. hom:clock qol:hasValue ?t.
 ?t func:hours-from-dateTime ?h.
 ?h math:notLessThan 22. ?h math:lessThan 7
} => {hom:runAway :getScore 9}.
{?u qol:detectedIn ?o. ?o a qol:Outside. ?u qol:inRoomFor ?d.
?d math:notLessThan 18000
} => {hom:runAway :getScore 9}.
```

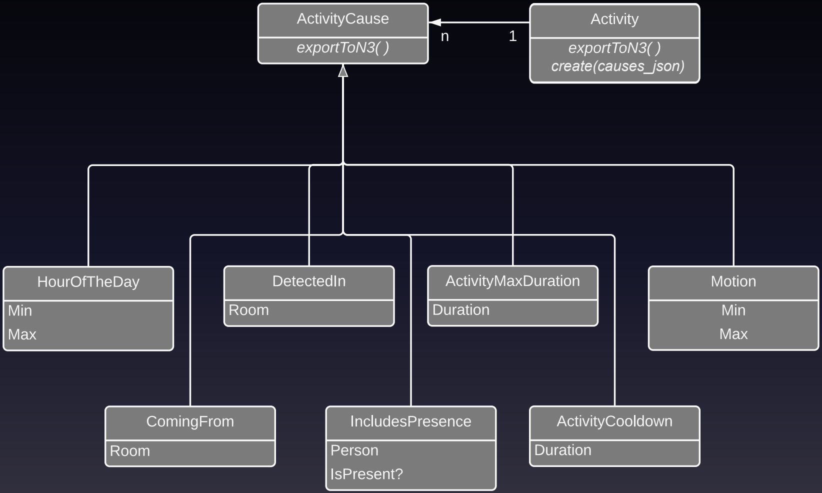
- …to that:

```
[{
  "name": "Run away",
  "conditions": {
    "hour_of_the_day": [22, 7],
    "detected_in": {"room": "room:Outside"},
    "min_duration": "5min"
  }
}]
```

Benefits:

- More standardized;
- Reduced redundancy;
- Errors are raised on compilation, not execution.

Making Rules Less Error-Prone



Rules Factory — The model

Knot — A Prototyping Framework

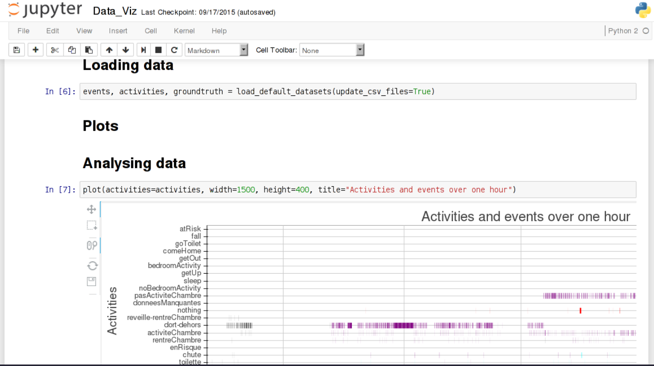
A brief overview…

- *Rules factory* & houses factory;
- Modular *reasoning engine*;
    - Tracing changes in the ontology;
- Data *preprocessing*;
- Data *visualization*;
- Realistic data generation;

Benefits

- Faster prototyping cycles;
- All-in-one environment;
- Rules can be exported into N3 format.

Knot — A Prototyping Framework

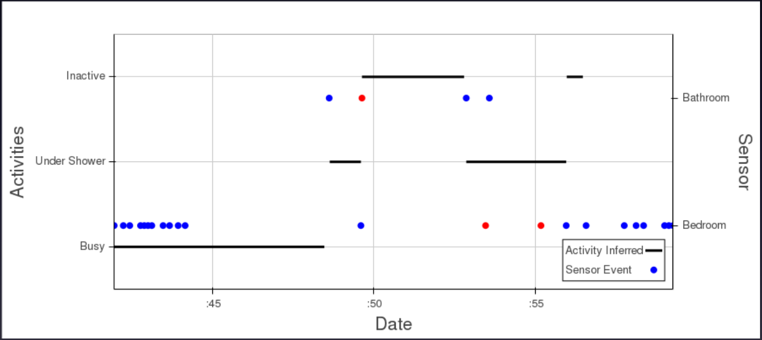


Knot's Integrated Development Environment

Outline

Problems Observed — 2/2

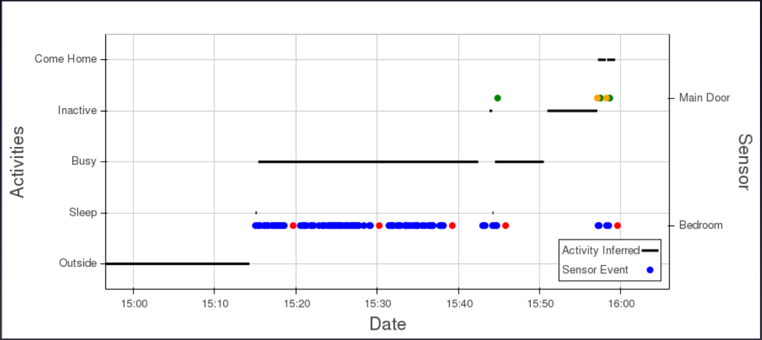
- ✓ Faulty inference due to a poorly defined rule;
- ➡ Reasoner trying to be too fine-grained:



- ➡ Reasoner being too coarse-grained;

Problems Observed — 2/2

- ✓ Faulty inference due to a poorly defined rule;
- ➡ Reasoner trying to be too fine-grained;
- ➡ Reasoner being too coarse-grained:



Introducing Hierarchical Activities

You could be *“in the kitchen”*, *“cooking”* AND *“cooking pasta”*, at the same time.

Introducing Hierarchical Activities

You could be *“in the kitchen”*, *“cooking”* AND *“cooking pasta”*, at the same time.

- Activities
  - Kitchen Activity
    - Eat Meal
    - Cook Meal
  - Activity in Bathroom
    - Go Toilet
    - Hygiene
  - Living Room Activity
    - Read Book
    - Watch TV
  - Bedroom Activity
  - Sleep
  - Outside
  - Come Home

Representation

```
:cookMeal a :Activity ;
   rdfs:label "Cook Meal"@en ;
   :generalizes :cookPasta .
:cookPasta a :Activity ;
   rdfs:label "Cook Pasta"@en ;
   :specializes :cookMeal .
```

Introducing Hierarchical Activities

You could be *"in the kitchen"*, *"cooking"* AND *"cooking pasta"*, at the same time.

Activities
- Kitchen Activity
  - Eat Meal
  - Cook Meal
- Activity in Bathroom
  - Go Toilet
  - Hygiene
- Living Room Activity
  - Read Book
  - Watch TV
- Bedroom Activity
- Sleep
- Outside
- Come Home

Representation

```
:cookMeal a :Activity ;
   rdfs:label "Cook Meal"@en ;
   :generalizes :cookPasta .
:cookPasta a :Activity ;
   rdfs:label "Cook Pasta"@en ;
   :specializes :cookMeal .
```

- A prerequisite for the proposed method of improving accuracy.

Introducing Granularity and Accuracy

- Precision :
  - *The amount of details attached to an activity.*
- Accuracy;
  - *The confidence of having a correct inference.*

- Given a hierarchy of activities,
  a child activity is always **more fine-grained** and **less accurate** than its parent.

Introducing Granularity and Accuracy

- ~~Precision~~ Granularity:
    - *The amount of details attached to an activity.*
- Accuracy;
    - *The confidence of having a correct inference.*

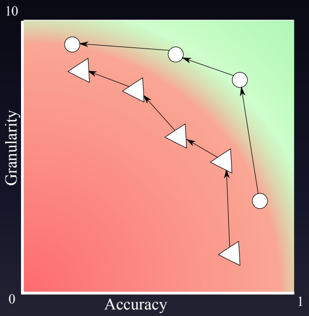
- Given a hierarchy of activities,
  a child activity is always **more fine-grained** and **less accurate** than its parent.

Introducing Granularity and Accuracy

- ~~Precision~~ Granularity:
  - *The amount of details attached to an activity.*
- Accuracy;
  - *The confidence of having a correct inference.*

- Given a hierarchy of activities,
  a child activity is always **more fine-grained** and **less accurate** than its parent.

Introducing Granularity and Accuracy



- Given a hierarchy of activities,
  a child activity is always **more fine-grained** and **less accurate** than its parent.

Measuring the Accuracy of an Activity

1. Observe the ground-truth, from a real deployment;
2. Simulate the activities in a living lab.

Measuring the Accuracy of an Activity

1. Observe the ground-truth, from a real deployment:
   - $P(A = a|I = a)$:
     - *"is the person really doing a, when the reasoner says a?"*

$$P(A = a|I = i) = \frac{|a \cap i| \sum_{X \in I} duration(X)}{|A \cap I| \sum_{x \in i} duration(x)} \tag{1}$$

where:

$|A \cap I|$:observations made while inferring an activity

$|a \cap i|$:occurrences when i is inferred and a is observed

$\sum_{x \in i} duration(x)$: total duration when i is inferred

$\sum_{X \in I} duration(X)$: total duration covered by our inferences

2. Simulate the activities in a living lab.

Measuring the Accuracy of an Activity

1. Observe the ground-truth, from a real deployment;
2. Simulate the activities in a living lab:

$$P(A = a | I = i) = \frac{|a \cap i| X}{|A \cap I| x} \tag{2}$$

where:

$|A \cap I|$:observations made while inferring an activity

$|a \cap i|$:occurrences when i is inferred and a is observed

x: occurrences when i is inferred

X: total number of inferences

A More Effective Method For Activity Recognition

- Looking for the activity with the best balance of accuracy and granularity.
  1. Assign a score to each activity, based on its accuracy and granularity.

$$Score(a) = \left(accuracy\ (a)^{A} \times (0.1 \times granularity\ (a))^{G}\right)^{\frac{1}{A \times G}} \tag{3}$$

Where:

A is the weight given to Accuracy

G is the weight given to Granularity

The 0.1 factor is used to normalize Granularity as a [0, 1] value

  2. When several activities are inferred at once, Select the activity with the highest score.

A More Effective Method For Activity Recognition

- Looking for the activity with the best balance of accuracy and granularity.
  1. Assign a score to each activity, based on its accuracy and granularity:

$$Score(a) = \left(accuracy\ (a)^{A} \times (0.1 \times granularity\ (a))^{G}\right)^{\frac{1}{A \times G}} \tag{3}$$

Where:

A is the weight given to Accuracy

G is the weight given to Granularity

The 0.1 factor is used to normalize Granularity as a [0, 1] value

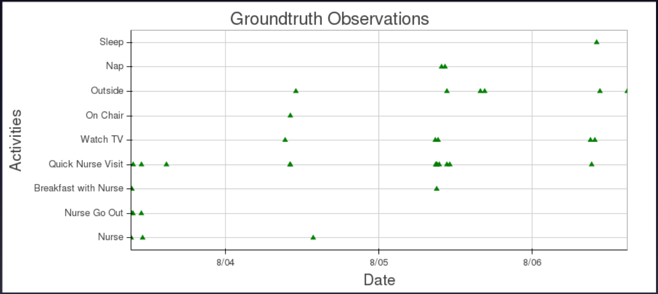
  2. When several activities are inferred at once, Select the activity with the highest score.

Remark: *The application of this method is reasoner-agnostic*

Outline

Validation — Observing Ground-Truth

- 465 observations in nursing home, over 10 days.



Validation — Introducing Metrics

- $T(A)$ — Measured accuracy of the reasoning
- $\bar{A}(activities)$ — Average value of accuracy
- $\bar{G}(activities)$ — Average value of granularity
- $\bar{S}(activities)$ — Average value of score

Validation — Introducing Metrics

- $T(A)$ — Measured accuracy of the reasoning:

$$T(A) = \frac{|\{groundtruth(a) = inferred(a) | a \in A\}|}{|groundtruth(A) \cap inferred(A)|} \qquad (4)$$

- $\bar{A}(activities)$ — Average value of accuracy
- $\bar{G}(activities)$ — Average value of granularity
- $\bar{S}(activities)$ — Average value of score

Validation — Introducing Metrics

- $T(A)$ — Measured accuracy of the reasoning
- $\bar{A}(activities)$ — Average value of accuracy:

$$\bar{A}(activities) = \frac{\sum\limits_{a \in activities} (accuracy\,(a) \times duration\,(a))}{\sum\limits_{a \in activities} duration\,(a)} \tag{5}$$

- $\bar{G}(activities)$ — Average value of granularity
- $\bar{S}(activities)$ — Average value of score

Validation — Introducing Metrics

- $T(A)$ — Measured accuracy of the reasoning
- $\bar{A}(activities)$ — Average value of accuracy
- $\bar{G}(activities)$ — Average value of granularity:

$$\bar{G}(activities) = \frac{\sum\limits_{a \in activities} (granularity\,(a) \times duration\,(a))}{\sum\limits_{a \in activities} duration\,(a)} \tag{6}$$

- $\bar{S}(activities)$ — Average value of score

Validation — Introducing Metrics

- $T(A)$ — Measured accuracy of the reasoning
- $\bar{A}(activities)$ — Average value of accuracy
- $\bar{G}(activities)$ — Average value of granularity
- $\bar{S}(activities)$ — Average value of score:

$$\bar{S}(activities) = \frac{\sum\limits_{a \in activities} (score(a) \times duration(a))}{\sum\limits_{a \in activities} duration(a)} \tag{7}$$

Validation — Results

- **48%** improvement of measured accuracy;
  - **38%** loss in granularity.

Sample	A	P	Total Measured Accuracy $T(A)$	Average Accuracy $\bar{A}$	Average Granularity $\bar{G}$	Average Score $\bar{S}$
Before calibration	*3*	*1*	63.4%	55.5%	6.75	56.9%
After calibration			93.8%	89.4%	4.15	73.4%
Before calibration	*1*	*1*	63.4%	67.3%	5.56	59%
After calibration			66.7%	55.5%	5.82	60.7%

Results with $(A = 3, G = 1)$ and $(A = 1, G = 1)$, before and after the calibration

Outline

Conclusion

Contributions:

- *UbiSMART*, a framework for Ambient Assisted Living:
  - User-friendly services;
  - Simplified setup, can be carried by a non-technician;
  - Automatic configuration per-house.
- A novel method to infer activities with an adequate granularity and accuracy;
- A prototyping engine for activity recognition.
  - Including a *"rules factory"*
- And also…
  - An algorithm to filter multi-user situations;
  - Two methods to detect inconsistencies in a rule engine.

Conclusion

Contributions:

- *UbiSMART*, a framework for Ambient Assisted Living:
  - User-friendly services;
  - Simplified setup, can be carried by a non-technician;
  - Automatic configuration per-house.
- A novel method to infer activities with an adequate granularity and accuracy;
- A prototyping engine for activity recognition.
  - Including a *“rules factory”*
- And also…
  - An algorithm to filter multi-user situations;
  - Two methods to detect inconsistencies in a rule engine.

Conclusion

Contributions:

- *UbiSMART*, a framework for Ambient Assisted Living:
  - User-friendly services;
  - Simplified setup, can be carried by a non-technician;
  - Automatic configuration per-house.
- A novel method to infer activities with an adequate granularity and accuracy;
- A prototyping engine for activity recognition.
  - Including a *“rules factory”*
- And also…
  - An algorithm to filter multi-user situations;
  - Two methods to detect inconsistencies in a rule engine.

Conclusion

Contributions:

- *UbiSMART*, a framework for Ambient Assisted Living:
  - User-friendly services;
  - Simplified setup, can be carried by a non-technician;
  - Automatic configuration per-house.
- A novel method to infer activities with an adequate granularity and accuracy;
- A prototyping engine for activity recognition.
  - Including a *“rules factory”*
- And also…
  - An algorithm to filter multi-user situations;
  - Two methods to detect inconsistencies in a rule engine.

Conclusion

Validations:

- A year of deployment in several real-world environment;
  - Including more than 300,000 recorded sensor events.
- Part of the “Chaire Quality of Life” and “VHP interactive” projects;
- Foundation for the *City4Age* Horizon2020 European project.

Perspectives

- Remaining challenges towards industrial large-scale deployment:
  - More accurate activity recognition;
  - More powerful service delivery:
    - Upcoming PhD thesis of Joaquim Bellmunt Montoya.
  - More adequate selection of sensors:
    - Upcoming PhD thesis of Ibrahim Sadek.
  - Improve performances for the reasoning engine.
- Opportunities:
  - Tracking the long-term evolution of elderly people's health;
    - Upcoming PhD thesis of Firas Kaddachi.
  - Towards smart-cities.

Thank you very much for your attention.

Any questions?