



**HAL**  
open science

# Modelling and Recognition of Human Activities of Daily Living in a Smart Home

Kévin Viard

► **To cite this version:**

Kévin Viard. Modelling and Recognition of Human Activities of Daily Living in a Smart Home. Automatic Control Engineering. Université Paris Saclay (COMUE); Politecnico di Bari. Dipartimento di Ingegneria Elettrica e dell'Informazione (Italia), 2018. English. NNT: 2018SACLN022. tel-01867623

**HAL Id: tel-01867623**

**<https://theses.hal.science/tel-01867623>**

Submitted on 4 Sep 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Modelling and Recognition of Human Activities of Daily Living in a Smart Home

Modélisation et reconnaissance des habitudes de vie  
d'un habitant dans un habitat intelligent

Modellizzazione e riconoscimento delle abitudini di un  
abitante in una casa intelligente

PhD Thesis of the Université Paris-Saclay and of the Politecnico  
di Bari, prepared at the LURPA (ENS Paris-Saclay) and at the  
LCA (Politecnico di Bari)

École doctorale n°580 Sciences et Technologies de l'Information  
et de la Communication (STIC) spécialité Automatique.  
Ph.D. School programm Ingegneria Elettrica e dell'Informazione  
XXXI ciclo.

Ph.D. defended on July 13 by

**Kévin VIARD**

## Committee :

<b>Alessandro Giua</b> Professor, University of Cagliari	Reviewer
<b>Jean-François Pétin</b> Professor, Université de Lorraine	Reviewer
<b>Mounir Mokhtari</b> Professor, Institut Mines-Télécom	President
<b>Maria Pia Fanti</b> Professor, Politecnico di Bari	Advisor
<b>Jean-Jacques Lesage</b> Professor, Ecole Normale Supérieure Paris-Saclay	Advisor
<b>Gregory Faraut</b> Ass. Professor, Ecole Normale Supérieure Paris-Saclay	Co-advisor



*"Do. Or do not. There is no try."*  
- Jedi Grand Master Yoda -  
George Lucas, *The Empire Strikes Back*



# Acknowledgments

---

In order to make my acknowledgments understandable by the concerned people, I will use the french and italian language in this page.

Tout d'abord, merci à Jean-Jacques pour les nombreux échanges que nous avons eus tout au court de cette thèse. J'ai pas mal parlé du côté "coup de bâton" le jour de ma soutenance, car ça n'a pas toujours été facile, mais il est important de préciser que la plupart des réunions se sont quand même très bien passées et que beaucoup de bonnes choses en sont sorties ! Ensuite, merci Gregory, pour ta capacité d'écoute et de soutien. Je ne compte plus les fois où j'ai dû te dire quelque chose du genre "ça me saoule, si ça continue comme ça, je vais tous les envoyer bouler et ça réglera le problème". À chaque fois tu m'as écouté et tu m'as ramené vers la voie de la diplomatie qui, admettons-le, est souvent la meilleure (et souvent la plus réfléchie).

Grazie Maria Pia per l'ascolto e l'aiuto che me ha dato e ti ringrazio per la fiducia. Grazie anche perché me ha dato la liberta di sviluppare i miei idee. Gracie a tuto il LCA per la buona accoglienza e la vostra gentilezza.

Merci à l'ensemble des doctorants du LURPA qui, ensemble, ont réussi à me donner envie de me lever tous les matins. L'ambiance entre doctorants est géniale au labo : gardez bien ça. Merci au CIVIL et à ses trois chefs incontestés que j'ai connus en temps que doctorant. J'ai toujours été invité aux soirées même si je ne venais presque jamais, et ça faisait plaisir. Merci Laureen pour le soutien et ton écoute. Merci Yannick d'avoir rendu la moitié de mes semaines de thèse plus heureuses. Merci Florian pour ta naïveté qui, bien que l'on s'en moque, fait quand même toujours plaisir et pour ta capacité à encaisser les "free fions" avec le sourire sans jamais te vexer. Merci aussi pour ces discussions endiablées que nous avons eues sur la N20 en refaisant le monde.

Merci à ma famille d'avoir toujours cru en moi et de m'avoir soutenu jusque là. D'avoir été présente, physiquement et/ou mentalement à ma soutenance. Ils ont tendance à dire: "tu as fait ça tout seul", mais je tiens à redire que non, ils m'ont mener jusque là. Merci à ma belle-famille également pour son soutien.

Enfin, merci à Constance, qui partage ma vie. Tu m'as récupéré démotivé plus d'une fois et tu m'as à chaque fois redonné la motivation de continuer. Tu m'as toujours soutenu sans jamais ciller, que ce soit lorsque j'ai passé l'agrégation ou tout au long de cette thèse. Tu as relu mon manuscrit de thèse (sûrement plus que ce que j'ai osé le relire moi-même) et tu m'as guidé vers l'aboutissement de celui-ci. Sans toi, je ne serai certainement pas en train d'écrire un mail de remerciement lié à l'aboutissement de ma thèse. Merci d'avoir tenu durant ces longs mois de séparations qu'implique le principe même de thèse en cotutelle. Enfin, merci d'avoir choisi de passer ta vie avec moi, d'être qui tu es, et de continuer notre aventure vers Clermont-Ferrand et vers une vie plus posée.

Merci à tous ceux que j'ai sûrement oublié, mais que j'aime quand même :D



# Contents

Contents	i
List of figures	iii
List of tables	vii
Frequently used abbreviations	ix
General Introduction	1
<b>1 Activities of Daily Living Issues and Objectives</b>	<b>5</b>
Introduction	6
1.1 Literature review	6
1.1.1 Terminology and definitions	6
1.1.2 Smart home technologies	8
1.1.3 Activities of daily living definition and main topics	11
1.1.4 Activity discovery	13
1.1.5 Activity recognition	16
1.1.6 Summary	19
1.2 Problem statement	20
1.2.1 Global objectives	20
1.2.2 Considered assumptions	21
1.2.3 Proposed framework to discover and recognise ADLs	22
Conclusion	24
<b>2 Case Study</b>	<b>25</b>
Introduction	26
2.1 Existing living labs and public datasets	26
2.1.1 Living lab of the Washington State University (WSU)	26
2.1.2 Living lab of the Université de Sherbrooke (Domus)	29
2.1.3 Living lab of the Massachusetts Institute of Technology	31
Conclusion	34
2.2 Ambient Assisted Living Test Area (AALTA)	35
2.2.1 Description	35
2.2.2 Activities to monitor	38
2.2.3 Sensor placement	39
2.2.4 Experimental protocol	42
Conclusion	44
<b>3 Activity Discovery</b>	<b>45</b>
Introduction	46



---

3.1	Models and notations . . . . .	46
3.2	A systematic procedure for models generation . . . . .	49
3.2.1	Generation of PFA structure . . . . .	49
3.2.2	Database of event logs exploration . . . . .	53
3.2.3	Probabilities computation . . . . .	55
3.3	Application to the Case Study . . . . .	61
3.3.1	Generation of PFA structure . . . . .	61
3.3.2	Database of event logs exploration . . . . .	63
3.3.3	Probabilities computation . . . . .	67
3.4	Discussion . . . . .	73
	Conclusion . . . . .	73
<b>4</b>	<b>Activity Recognition</b>	<b>75</b>
	Introduction . . . . .	76
4.1	Models and used notations . . . . .	76
4.2	Activity Recognition protocol . . . . .	77
4.2.1	Observed sequence windowing . . . . .	77
4.2.2	Projection of the exploited sequence . . . . .	78
4.2.3	Language creation using the projected sequence . . . . .	79
4.2.4	Probability estimation . . . . .	81
4.3	Distance between a sequence and a PFA . . . . .	82
4.3.1	The likelihood . . . . .	82
4.3.2	The perplexity . . . . .	83
4.3.3	Issues due to distance hypotheses . . . . .	84
4.4	Normalised likelihood and extension . . . . .	85
4.4.1	The normalised likelihood . . . . .	85
4.4.2	The normalised perplexity . . . . .	86
4.4.3	Computational Issues . . . . .	87
4.5	Application to the Case Study . . . . .	96
4.5.1	Observed sequence windowing . . . . .	96
4.5.2	Projection of the exploited sequence . . . . .	96
4.5.3	Language generation based on the projected sequence . . . . .	97
4.5.4	Probability estimation . . . . .	100
4.6	Discussion . . . . .	102
4.6.1	Activities occurring individually . . . . .	102
4.6.2	Activities occurring successively . . . . .	104
4.6.3	Validation . . . . .	105
	Conclusion . . . . .	105
	<b>Conclusions &amp; Outlooks</b>	<b>107</b>
	<b>Bibliography</b>	<b>111</b>
	<b>A Proofs</b>	<b>123</b>
	<b>B Generation of the activities hot beverage preparation and use bath- room PFA structures</b>	<b>127</b>
	<b>C Counting example during the activities hot beverage preparation and use bathroom performances.</b>	<b>131</b>

# List of Figures

0.0.1	European Union age repartition in 2010 and 2060 (EUROSTAT, 2010).	1
0.0.2	Population aged 60 and over in 2012(a) and 2050(b) (World Health Organization, 2012).	2
0.0.3	Population pyramid according to the United Nations World Population Prospects (United Nations, 2017).	2
0.0.4	Overview of the contributions of the thesis.	3
1.1.1	A smart home: a home equipped with several sensors and actuators.	7
1.1.2	(a) A demonstration of wearable body sensors on a human body. (b) A conceptual BSN architecture of the proposed AAL system. (c) An example of wearable textile sensors (Forkan et al., 2014).	9
1.1.3	Sensors taxonomy	11
1.1.4	(a) Human behaviour analysis tasks - classification (Chaarouai et al., 2012) (b) Human behaviour degree of semantics (Cook and Krishnan, 2015).	13
1.1.5	DBN representation for a standard HSMM. Shaded nodes represent observation. (Duong et al., 2009).	14
1.1.6	General data flow for training systems based on wearable sensors (Lara and Labrador, 2013).	15
1.1.7	Generic data acquisition architecture for Human Activity Discovery and Recognition (Lara and Labrador, 2013).	15
1.1.8	Context model using OWL (Forkan et al., 2014).	17
1.1.9	Thesis summarised context	19
1.1.10	Existing methods, their sensor level of semantics and their intrusiveness.	20
1.2.1	Activity discovery and recognition proposed framework.	23
1.2.2	$I_{AD}^1$ : Hierarchical decomposition of activities into actions linked to sensor events given by an expert.	23
2.1.1	Resident performing “hand washing” activity (left). This activity triggers motion sensor ON/OFF events as well as water flow sensor values (right). Sensors in the apartment (bottom): monitor motion (M), temperature (T), water (W), burner (B), phone (P), and item use (I) (Cook et al., 2009).	27
2.1.2	Example of logs extracted from the CASAS file "p01.t1".	29
2.1.3	Domus smart home plan (Chikhaoui et al., 2010).	30
2.1.4	Example of logs extracted from the file "Domus Series 1/User 1/Day 1.vna".	31
2.1.5	(a) The top view of the apartments and the sensor distribution for subject one. (b) Sensor distribution for subject two. Tapia et al. (2004).	32
2.1.6	Exemple of logs extracted from the Tapia et al. (2004) file "activities_data.csv" of the "subject2" directory.	33
2.1.7	Example of readable presentation of Tapia et al. (2004) logs.	34
2.2.1	The smart flat with its zones	35
2.2.2	Picture of the AALTA entrance (a) and bathroom (b).	36

2.2.3	Picture of the kitchen (before equipping sensors and machines). . . . .	37
2.2.4	Picture of the sleeping zone. . . . .	37
2.2.5	Hierarchical links between our activities and our actions. . . . .	38
2.2.6	Event emission from sensor binary information. . . . .	39
2.2.7	Installed kinds of binary sensors. . . . .	40
2.2.8	Kitchen view of the smart flat. . . . .	41
2.2.9	The smart flat with its binary sensors . . . . .	41
2.2.10	Expert hierarchical decomposition of activities to monitor . . . . .	42
2.2.11	Living lab sensor networks . . . . .	42
2.2.12	Example of AALTA logs. . . . .	43
2.2.13	Structure of the test sequence of observed events . . . . .	44
3.1.1	Graphical representation of the hierarchical decomposition of activities into actions linked to sensor events given by an expert. . . . .	47
3.1.2	Graphical representation of a PFA $\mathcal{A}_k$ (Vidal et al., 2005b). . . . .	48
3.2.1	Example of the creation of a set of five states $Q_{\mathcal{A}_k} = \{q_0, q_1, q_2, q_3, q_4\}$ from an activity $\mathcal{A}_k$ decomposed into four actions (Action $k.1$ to $k.4$ ). . . . .	50
3.2.2	Example of the creation of five transitions $\{(q_0, e_4, q_3), (q_1, e_4, q_3), (q_2, e_4, q_3), (q_3, e_4, q_3), (q_4, e_4, q_3)\}$ reflecting the link between the action $N.3$ and the event $e_4$ . . . . .	50
3.2.3	From expert decomposition to PFA structure. . . . .	52
3.2.4	Common approaches for processing streaming data (from (Krishnan and Cook, 2014)) . . . . .	53
3.2.5	Sliding sensor window and indicator computation . . . . .	57
3.3.1	Expert hierarchical decomposition of activities to monitor and merge groups $Me_l$ for representation . . . . .	61
3.3.2	(a) Developed representation; (b) Merged representation to improve readability. . . . .	62
3.3.3	Generation of the activity $\mathcal{A}_1$ (Cooking) PFA structure. . . . .	63
3.3.4	Final $\mathcal{A}_1$ <i>Cooking</i> model. . . . .	70
3.3.5	Final $\mathcal{A}_2$ <i>Hot beverage preparation</i> model. . . . .	71
3.3.6	Final $\mathcal{A}_3$ <i>Use bathroom</i> model. . . . .	72
4.2.1	The proposed activity recognition protocol . . . . .	77
4.2.2	Chosen language generation: language of substrings of length upper than two. . . . .	81
4.4.1	Example on a simple PFA $\mathcal{A}_k$ . . . . .	86
4.4.2	<i>Reduced PFA</i> associated with the PFA $\mathcal{A}_1$ (figure 3.3.4). . . . .	89
4.4.3	Computation of the maximum likelihood without dynamical computation. . . . .	93
4.4.4	Computation of the maximum likelihood with dynamical computation . . . . .	93
4.5.1	Inverse of the normalised perplexity during the realisation of the activity $\mathcal{A}_1$ cooking using no shared events. . . . .	101
4.6.1	Inverse of the normalised perplexity during the realisation of the activity $\mathcal{A}_3$ : <i>use bathroom</i> sharing no event with any other activity. . . . .	103
4.6.2	Inverse of the normalised perplexity during the realisation of the activity $\mathcal{A}_2$ : <i>hot beverage preparation</i> sharing some events with the activity $\mathcal{A}_1$ : <i>cooking</i> . . . . .	103

---

4.6.3	Inverse of the normalised perplexity during the realisation of the activity $\mathcal{A}_1$ : <i>cooking</i> sharing many events with the activity $\mathcal{A}_2$ : <i>hot beverage preparation</i> . . . . .	104
4.6.4	Inverse of the normalised perplexity during the succession of the three activities $\mathcal{A}_1$ : <i>cooking</i> , $\mathcal{A}_2$ : <i>hot beverage preparation</i> and $\mathcal{A}_3$ : <i>use bathroom</i> sharing no zero of some events. . . . .	105
Fr.1	Structure de la population par tranche d'âge, UE-28, 2015-2080 (en % de la population totale) (EUROSTAT, 2013). . . . .	117
Fr.2	Pourcentage de la population de tout âge ayant un excellent ou un très bon état de santé, par âge et par sexe : États-Unis (Norris et al., 2017). . . . .	118
Fr.3	Plan proposé pour la découverte et la reconnaissance d'activités. . . . .	119
B.0.1	Generation of the activity $\mathcal{A}_2$ (Hot beverage preparation) PFA structure. . . . .	128
B.0.2	Generation of the activity $\mathcal{A}_3$ (Use bathroom) PFA structure. . . . .	129



# List of Tables

1.1.1	Examples of some typical body sensors and their use in AAL (Forkan et al., 2014). . . . .	9
1.1.2	Review of existing sensors . . . . .	12
1.1.3	Service rules for detecting a possible heart attack (Forkan et al., 2014). . . . .	17
2.2.1	List of events logged in AALTA. . . . .	40
3.3.1	Counting example during the activity $\mathcal{A}_1$ performance. . . . .	65
3.3.2	$N_{e_i \rightarrow e_j}^1$ values for the activity $\mathcal{A}_1$ , with $e_i$ in column and $e_j$ in line (e.g. $N_{e_{13} \rightarrow e_{24}}^1 = 7$ ). . . . .	66
3.3.3	$N_{init\ e_i}^1$ values for the activity $\mathcal{A}_1$ (e.g. $N_{init\ e_{13}}^1 = 35$ ). . . . .	66
3.3.4	$N_{e_i}^1$ values for the activity $\mathcal{A}_1$ (e.g. $N_{e_{13}}^1 = 132$ ). . . . .	66
4.4.1	Time complexity of the maximum classical likelihood computation for a given length of sequence equal to $ w^{Obs} $ with $card(Q_{\mathcal{A}_k}) \ll card(\Sigma_{\mathcal{A}_k})$ . . . . .	95
4.5.1	Analysed sequences each time a new event of the sequence $w_{\mathcal{A}_k}^{Obs}$ occurs. . . . .	97
4.5.2	Projected sequences $w_{p_k}^{Obs}$ when new events of the sequences $w_{\mathcal{A}_k}^{Obs}$ occurs. . . . .	98
4.5.3	Generated languages $\mathcal{L}_{p_k}^{Obs}$ when new events of the sequence $w_{\mathcal{A}_1}^{Obs}$ occurs. . . . .	98
4.5.4	Generated languages $\mathcal{L}_{p_k}^{Obs}$ when new events of the sequence $w_{\mathcal{A}_2}^{Obs}$ occurs. . . . .	99
4.5.5	Generated languages $\mathcal{L}_{p_k}^{Obs}$ when new events of the sequence $w_{\mathcal{A}_3}^{Obs}$ occurs. . . . .	99
4.5.6	Pertinent intermediate probability for the distance computation of the example observed sequence and final distance. . . . .	100
C.0.1	Counting example during the activity $\mathcal{A}_2$ performance. . . . .	131
C.0.2	Counting example during the activity $\mathcal{A}_3$ performance. . . . .	132



# Frequently used abbreviations

---

<b>AAL</b>	Ambient Assisted Living
<b>AALTA</b>	Ambient Assisted Living Test Area
<b>AD</b>	Activity Discovery
<b>ADL</b>	Activity of Daily Living
<b>AR</b>	Activity Recognition
<b>HaH</b>	Health at Home
<b>PFA</b>	Probabilistic Finite-State Automata





# General Introduction

*"Quand même c'est classe, les vieux. Ils arrivent, c'est hyper mystérieux."*  
- Perceval le Gallois -  
Alexandre Astier, *Kaamelott, Saison 3, Episode 13 : La Poétique - 2ère partie*

## Motivation

Population dependency rate (i.e. the ratio of dependent people over the total population) has continuously increased over the last decades all over the world and is projected to continue to increase in the future. In fact, as shown in [EUROSTAT \(2010\)](#) and [Figure 0.0.1](#), the age repartition in the European Union population is drastically changing towards elder median age. In these societal studies, a person is considered as dependent on the rest of the population if its age is lower than 14 years old or upper than 65.

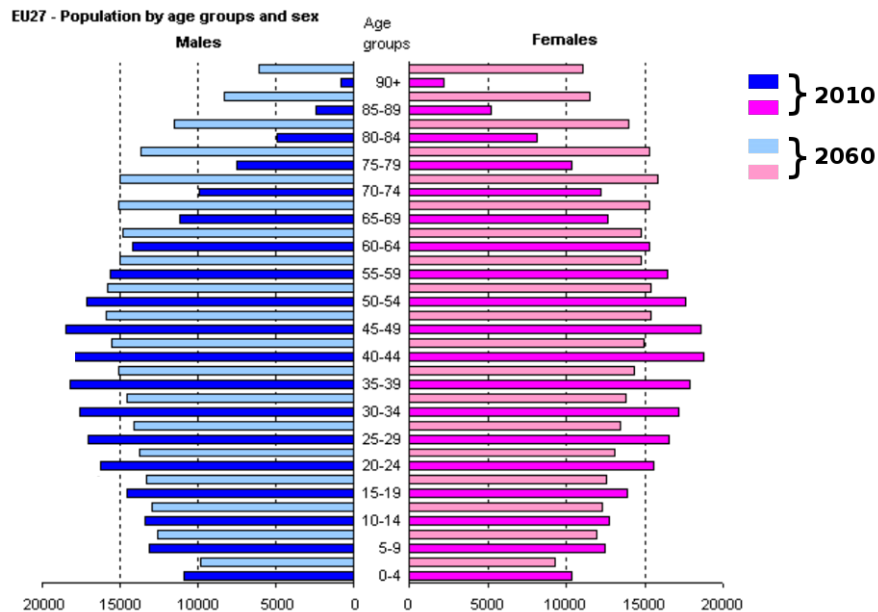


Figure 0.0.1: European Union age repartition in 2010 and 2060 ([EUROSTAT, 2010](#)).

As might be expectable, this population aging issue is not limited to the European Union. According to the [World Health Organization \(2012\)](#), [Figure 0.0.2](#), in 2050, the part of the population aged 60 or more will rise to 30% in the majority of countries. The United Nations world population prospects realised in 2017 supports this point, see [figure 0.0.3](#). In fact, the world population pyramid expected in 2050 shows a dependency rate drastically higher in the future.

This societal evolution is becoming an important human and economic issue for next years. Indeed, current health and well-fare institutions will not be sufficient to treat

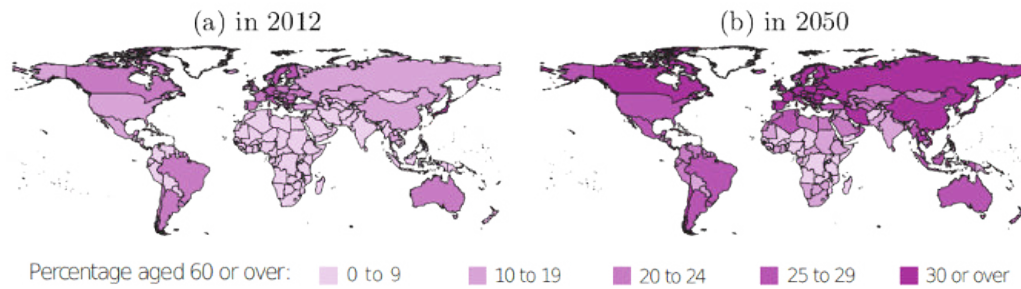


Figure 0.0.2: Population aged 60 and over in 2012(a) and 2050(b) (World Health Organization, 2012).

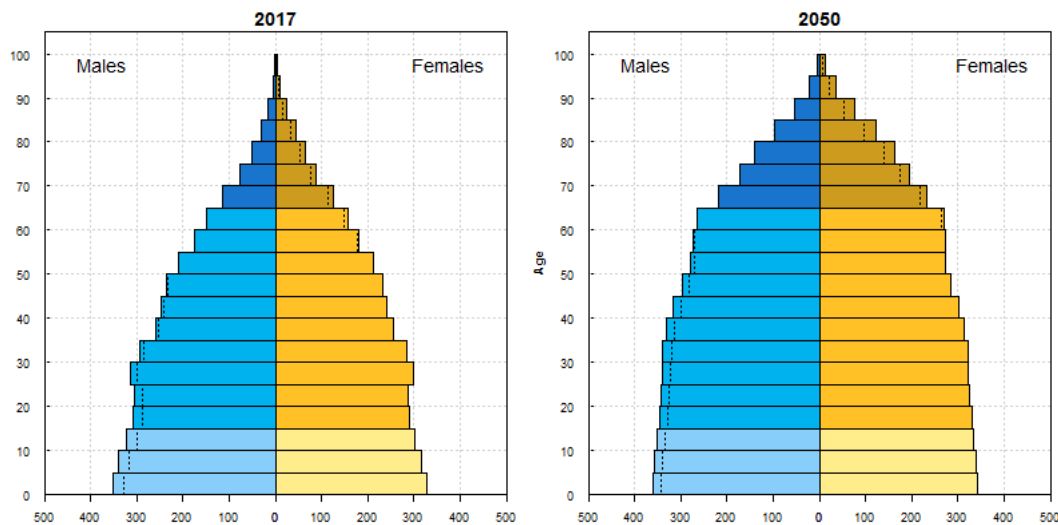


Figure 0.0.3: Population pyramid according to the United Nations World Population Prospects (United Nations, 2017).

this proportion of elderly people. Therefore, alternative solutions have to be found and rapidly developed in order to supply help and independence to dependent people.

A common idea to deal with this future lack of medical facilities is the development of ambient assisted living (AAL) systems that enable to keep people at home. In this way, medical facilities could be allocated to severe pathologies and emergencies only. Furthermore, it enables the elderly and disabled people to stay longer in a familiar environment (Cornelis et al., 2017). Therefore this cheaper, and more pleasant solution, needs to be developed. AAL systems allocated to health monitoring, also called health at home (HaH) systems, consist of keeping old people at home as long as possible, thanks to an automatic monitoring of their everyday life. As soon as an unsafe or abnormal behaviour is detected, the medical staff in charge or the family is informed.

The activities of daily living (ADLs) monitoring (Chaaroui et al., 2012; Cook and Krishnan, 2015) is one of the major investigations in the domain of HaH systems. It aims to provide the medical staff with very useful and precise information concerning the monitored patient. By definition (Lawton and Brody, 1969), an ADL is an activity, which is daily performed by a person (e.g. to prepare meal, to do housework, to take

leisure, etc.) and whose monitoring is helpful to the doctor. Current studies about ADLs mainly treat four problems: the activity discovery (AD), the activity recognition (AR), the activity prediction (AP) and the detection of deviation (DD). In this thesis, only the AD and the AR are treated. The AD objective is to generate one or several more or less formal models of activities by studying the patient habits during a learning period. After that, the objective of AR is to detect that an activity is actually performed by a person during its realisation. At this point, the acceptance of the proposed system by the monitored person is a major issue. The system has to be considered as non-intrusive by the patient while giving pertinent information to the medical staff.

The objective of this thesis is to propose both an activity discovery and an activity recognition method compatible with some assumptions linked with the acceptance issue and possibility to feasibly apply them in mass. These assumptions are explained with more details in section 1.2.2.

## Contributions of the thesis

The contributions of the presented thesis are summarised in figure 0.0.4.

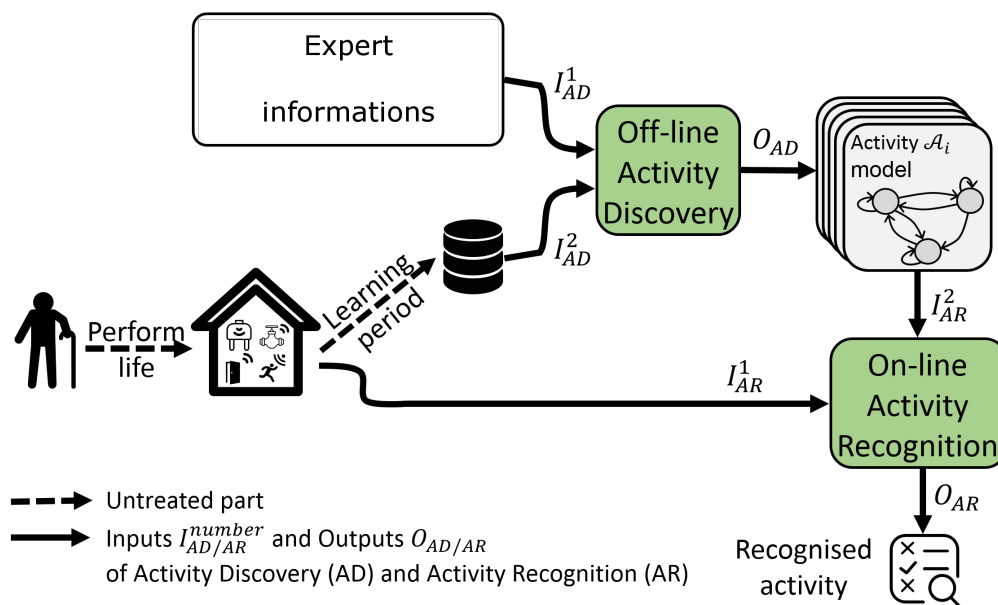


Figure 0.0.4: Overview of the contributions of the thesis.

In a first step, models of activities performed by an inhabitant in a smart home are built during an *Off-line Activity Discovery*. The presented AD method is performed using a database obtained by recording the events emitted by the instrumented home during a learning period. An additional expert knowledge is needed in order to produce efficient models. In the majority of existing works (Tapia et al., 2004; Gaglio et al., 2015), this expert knowledge consists of labelling of activities performed during the learning period. This expert knowledge is very difficult to obtain and error-prone. Therefore, in this thesis, we decided not to use this expert knowledge during the ADL study in order to keep our method applicable easily anywhere. Thus, in contrast with already existing methods, we reject the labelling of the performed activities during the learning period. In addition, to deal with the inhabitant willingness and privacy, we decide to use only

binary sensors.

For a second time, an *On-line Activity Recognition* protocol has been developed using the models previously computed with the AD to recognise which activity is performed by the inhabitant during its life living. As a consequence of the reject of the performed activity knowledge during our AD, the models obtained are not compatible with the already existing AR methods which have to be adapted. To this aim, new probabilistic distances and methods to compute them were defined.

## Organisation of the manuscript

The manuscript is organised as follows. A review of the state of the art in the fields of ambient assisted living and of activities of daily living monitoring is given in chapter 1. In this chapter, the concept of smart home, activity discovery and activity recognition are given and several existing technologies, sensors and methods are presented. At the end of this chapter, a problem statement is made and a summary of the assumptions and considerations hold in this thesis are enunciated and explained.

Before developing the two main contributions of this thesis, the case study we used to apply our methods is presented chapter 2. In this chapter, an overview of the existing living labs is first done and lacking information is exposed. Then, a description of the smart flat we developed for this thesis is presented.

In chapter 3, the first contribution consisting in an original Activity Discovery method is presented. An application of this method on our living lab is also proposed.

The second contribution of this thesis is presented in chapter 4. This contribution consists in a procedure to recognise which activity is performed by the inhabitant of a smart home. To this aim, some existing distances between an observation and a probabilistic model are presented and their limits are highlighted. Then, these distances are extended to be adapted to a more general case. To conclude the chapter 4, an application of the proposed activity recognition procedure in our case study is developed and discussed.

In the last chapter, a summary of the contributions of the thesis and outlooks for future works are given.

Finally, in the appendix of the thesis, the proofs of two properties enunciated in chapter 4 are extensively developed.

# Chapter 1

## Activities of Daily Living Issues and Objectives

---

*"Insanity is doing the same thing over and over again and expecting different results"*  
- Albert Einstein -

### Abstract

---

This chapter surveys the existing works and tools in the field of monitoring of activities of daily living, especially for activity discovery and activity recognition. A statement of the problem handled in this thesis is afterwards presented.

---

### Contents

---

Introduction . . . . .	6
1.1 Literature review . . . . .	6
1.1.1 Terminology and definitions . . . . .	6
1.1.2 Smart home technologies . . . . .	8
1.1.3 Activities of daily living definition and main topics . . . . .	11
1.1.4 Activity discovery . . . . .	13
1.1.5 Activity recognition . . . . .	16
1.1.6 Summary . . . . .	19
1.2 Problem statement . . . . .	20
1.2.1 Global objectives . . . . .	20
1.2.2 Considered assumptions . . . . .	21
1.2.3 Proposed framework to discover and recognise ADLs . . . . .	22
Conclusion . . . . .	24

---

## Introduction

As stated in the general introduction, due to the increase of the population median age, alternatives to classical medical facilities should be found to efficiently take care of dependent people. One of the most retained solutions is to maintain in their home environment the people who do not have too severe pathologies. To this aim, it is necessary to monitor these persons using data extracted from an adapted instrumentation. To achieve such a goal, different approaches based on different scientific backgrounds, different models and different considerations on the sensor that may be installed were developed.

This chapter is aimed to review the state of the art relative to this topic and to state the problem of this thesis. First, some definitions and generic terms are given and the concept of ambient assisted living (AAL) system and smart home are developed. Secondly, a technological description of the existing sensors are given and their cases of use are discussed. Third, a complete definition of Activities of Daily Living (ADLs) is related and the main topics around these ADLs are listed. The two topics developed in this thesis are thus presented in detailed and existing works are discussed.

Finally, the proposed framework to deal with activity discovery (AD) and recognition (AR) concludes this chapter.

### 1.1 Literature review

In this section, basic definitions of classical terms used in the domain of activities of daily living monitoring in a smart home are given. Technologies implied in smart home and their advantages and drawbacks are discussed. Then, existing works in AD and AR are described.

#### 1.1.1 Terminology and definitions

During the 20th century, a huge technological evolution was observed, especially in the fields of information and electronic networks. Compact sensors with low energy consumption and giving specific information were created. Naturally, with this technology evolution the notion of ubiquitous networking has appeared:

**Definition 1.1** (Ubiquitous networking). (*Rouse and Wigmore, 2017*) *Ubiquitous networking, also known as pervasive networking, is the distribution of communication infrastructures and wireless technologies throughout the environment to enable continuous connectivity. That capacity is an essential component of pervasive computing. (The terms are interchangeable, with slight variations, as either "ubiquitous" or "pervasive", which means essentially the same thing.)*

Several objectives can be reached by instrumenting the right environment. Indeed, ubiquitous environment could improve the energy consumption of a building (*Shah and Rabaey, 2002*), (*Agarwal et al., 2010*) or improve home safety (*Flöck, 2010*). Furthermore, ubiquitous environments can be used to manage the aging or disabled population life. Ambient assisted living has thus been envisaged monitoring the person life at any time. At this point the monitoring can be anywhere: at home, at work, in the supermarket...

**Definition 1.2** (Ambient Assisted Living (AAL)). (*Monekosso et al., 2015*) *Ambient assisted living can be defined as "the use of information and communication technologies in a person's daily living and working environment to enable them to stay active longer, remain socially connected and live independently into old age"*

By considering the case of health monitoring, the main environment of life for disabled and aging people is their own home. In the presented thesis, the goal is thus to monitor the people at home health. In this specific case, the terms *Health at Home systems* (HaH) or *Home Care Systems* (HCS) are used and defined as following:

**Definition 1.3** (Home Care Systems). (*Kleinberger et al., 2007*) *The aim of Home Care Systems is to allow the assisted persons to live longer in their preferred environment at home, while retaining their independence, even when they have handicaps or medical diseases.*

It is important to note here that challenges implied by HaH (or HCS) and their solutions are strongly dependent on technologies. It is at the same time worn by and limited by the existing technologies and their cost (*Barlow and Venables, 2003*). To develop HCS, it appears that the elderly or disabled people homes have to be equipped: the notion of smart home (*Robles and Kim, 2010*) is born.

**Definition 1.4** (Smart Home). (*Balta-Ozkan et al., 2014*) *A smart home is a residence equipped with a communication network, linking sensors, domestic appliances, and devices, that can be remotely monitored, accessed or controlled (*King, 2003*) and which provides services that respond to the needs of its inhabitants (*Chan et al., 2008; Taylor et al., 2007*). In principle, the term 'smart home' may refer to any form of residence, for example, a standalone house, an apartment, or a unit in a social housing development. In this definition, sensors are devices used to detect the location of people and objects, or to collect data about states (e.g. temperature, energy usage, open windows). Devices can be electronic, for instance, phones, televisions, computers, or electric, referring to the simplest toasters, kettles, light bulbs, etc.*

*The network, connecting and coordinating these various technological features (i.e. sensors, devices, appliances) and information, is central to the concept of the smart home (*King, 2003; Jiang et al., 2004*). It is the existence of this home network that distinguishes the smart home from a home merely equipped with standalone, highly advanced technological features (*Scott, 2007*).*

Figure 1.1.1 is a representation of a smart home in its basic definition: a home equipped with sensors and actuators connected via a communication network.



Figure 1.1.1: A smart home: a home equipped with several sensors and actuators.



The definitions of basic terms, from *ubiquitous environments* to *smart home* are now given. The description of technologies used in different smart homes is described in the next subsection.

### 1.1.2 Smart home technologies

The achievable objectives, and the ways to reach them, are strongly linked to smart home used technologies. In this subsection, sensors are divided considering the semantic level of information they give. We can define a three-bar ladder:

- High level of semantics,
- Average level of semantics,
- Low level of semantics.

#### High level of semantics

Some sensors give complex and complete information about the person to monitor. In this category, we classify wearable vital sign sensors (i) and cameras (ii).

- (i) Vital sign sensors, as used in [Forkan et al. \(2015\)](#) and [Sadek et al. \(2015\)](#), are wearable sensors, also called body sensors, which periodically send different vital signs data of the user to a cloud server. Table 1.1.1 shows some examples of typical body sensors. These kinds of sensors together form a body sensor network (BSN) ([Otto et al., 2006](#); [Bellifemine et al., 2011](#)). These sensors have some key configurations and infrastructure that make them easily implantable or wearable on the human body (Figure 1.1.2). Some sensors can be implanted in the garment of the user; these are known as wearable textile sensors. These sensors have low power, can communicate wirelessly and above all monitor the health and activity of the target user.
- (ii) Cameras are sensors giving information via image processing and motion recognition. Indeed, as precise in [Chaaroui et al. \(2012\)](#), motion recognition is the basis for estimation of human pose and gaze direction (also referred to as focus of attention) and for further human behaviour analysis tasks. Motion can be seen as a series of poses along the time. The human body is an articulated system of rigid segments connected by joints (as models used in [Andriluka et al. \(2009\)](#) and [Sapp et al. \(2010\)](#) assume). Human motion is often considered as a continuous evolution of the spatial configuration of these segments or body postures (as stated in [Li et al. \(2008\)](#) and exploited in [Andriluka et al. \(2009\)](#) and [Sapp et al. \(2010\)](#)). On the other hand, the gaze can either be seen as a line in the 3D space or a cone, or, if working only in the horizontal plane (as some works do, as seen later on), a direction and an angle.

#### Average level of semantics

In contrast with high level of semantics sensors, other sensors are giving a less complex but not poor information. It is the case of many wearable sensor not giving vital signs (i) or of microphones (ii).

- (i) Wearable sensors not giving vital signs are sensors partially worn by the inhabitant and giving binary information to the health at home system. It is the case of sensors based on the radio frequency identification (RFID) technologies. RFID is a technology used for identifying persons who carry identification badges (or tags)

Table 1.1.1: Examples of some typical body sensors and their use in AAL (Forkan et al., 2014).

Sensor	Measured signal	Application
ECG	Electrocardiogram wave	Heart rate
PPG	Photoplethysmogram wave	Blood volume pulse
BP	Blood pressure in mm Hg	Blood pressure
EEG	Electroencephalogram wave	Abnormality
EMG	Electromyograph wave	Muscular activity
Accelerometer	Acceleration in 3D space	Activity recognition
Motion sensor	Motion signal	User movement
Activity sensor	3-axis motion	Activity recognition
Inertial sensor	Motion signal	Position detection
BG sensor	Blood sugar level	Diabetes detection
Gyroscopes	Rotation angle	Body orientation
Thermometer	Body temperature in $^{\circ}F$	Fever detection
RF antenna	RF wave	Position detection
Fall detector	Motion signal	Fall detection

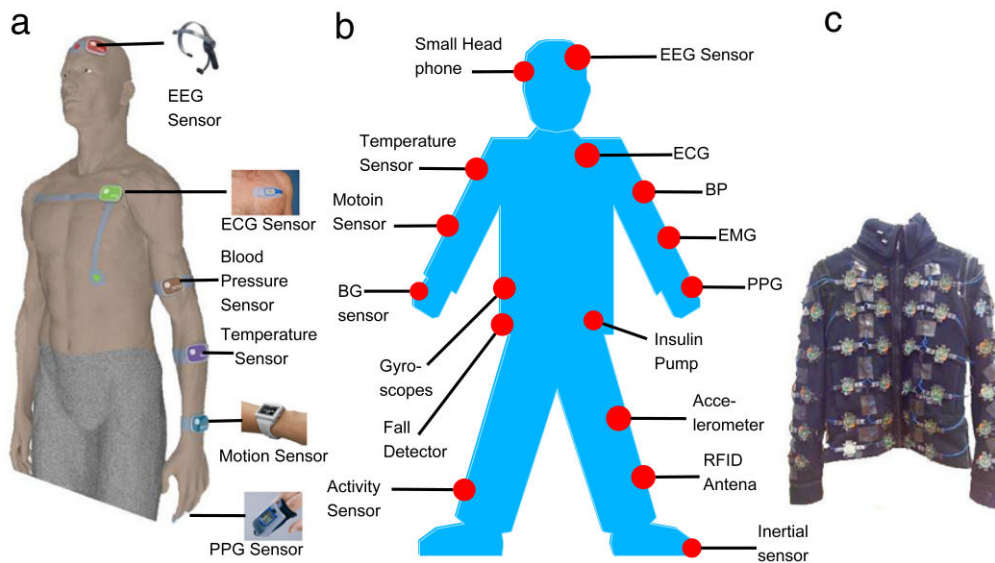


Figure 1.1.2: (a) A demonstration of wearable body sensors on a human body. (b) A conceptual BSN architecture of the proposed AAL system. (c) An example of wearable textile sensors (Forkan et al., 2014).

or a reader. If the person wears a tag (Hussain et al., 2009), this technology implies readers which read an approaching tag to identify the person who is carrying the tag. Else, if the person wears the reader (Darianian and Michael, 2008), it identifies the tagged objects or the tagged zones the person is approaching.

- (ii) Microphones are also used as sensors giving complementary information to other sensors as cameras (Brdiczka et al., 2009) or RFID (Park and Kautz, 2008).

### Low level of semantics

Finally, some sensor technologies provide a low level of semantics information. These sensors give binary values extracted from environmental sensing. As precise in [Cook and Krishnan \(2015\)](#), these binary environmental sensors are not affixed to the individuals performing the activity but are placed in the environment surrounding them. These sensors are valuable in passively providing readings without requiring individuals to comply with rules regarding wearing or carrying sensors in prescribed manners. The most frequent binary environmental sensors are:

- Passive infrared (PIR) sensors, or motion sensors, detect infrared radiations that are emitted by objects in their field of view through multiple slots. If the difference in the detected radiation between the multiple slots of a PIR sensor is greater than a predefined threshold (as would happen when a warm body moves into or out of the range of the sensor), it generates a message. A PIR sensor will sense movement from any object that generates heat, even if the origin is inorganic.
- Magnetic contact switch sensors, or magnetic door sensors, consist in two components: a reed switch and a magnet. When the door is closed, the magnet component pulls the metal switch in the second component closed so the electric circuit is complete, thus changing the state of the sensor. The sensor can report this change of state as a sensor event. When the magnet is moved by opening the door, the spring snaps the switch back into the open position. This cuts off the current and closes the relay, again causing a change in the state of the sensor that can be reported as a sensor event. This feature is useful for detecting if doors, windows, drawers, or cabinets are open or closed.
- Vibration sensors often attached to items or placed on surfaces in order to detect interactions with the corresponding object. Some sensors are designed to be sensitive both to vibration (dynamic acceleration) and tilt (static acceleration). While they can be useful for generating events when the object they are attached to is handled, they may also generate events when they are accidentally bumped or when nearby surfaces shake.
- Pressure sensors, for monitoring of activities in a particular environment. Tactile sensors are sensitive to touch, force, or pressure. These pressure sensors detect and measure interactions between an individual and a contact surface. The combined force can be compared to a threshold value to note that there is an object in contact with the sensor. Pressure sensors can be placed on or under chairs, door mats, floors, and beds to monitor the location and weight distribution of an individual in the space.
- Flow sensors provide readings indicating the amount of electricity or water that was consumed by a particular building for a unit of time. The metre calculates the amount of electricity or water currently being consumed and can report instantaneous values, accumulated values, or sufficient changes in consumption. The amount of electricity or water currently being consumed can be compared with a threshold value to note that an object is in use.
- Temperature sensors, light sensors, humidity sensors can be placed in environments to measure ambient temperature, lighting, and humidity. These types of sensors are frequently bundled into one package. Such sensors are calibrated to periodically report their current status (e.g., light level, humidity reading, and temperature reading) or they report their current reading when there is a sufficiently large change in the value from the previous time point.

From another point of view, it is possible to classify the sensor technologies by considering their interactions relatively to the human. In [Lara and Labrador \(2013\)](#), the sensors are classified in two families: sensors performing external or wearable sensing. Another classification, as proposed in [Chikhaoui et al. \(2011\)](#), orders events by considering the human feeling about the sensors in terms of privacy. Sensors are thus considered as intrusive or non-intrusive. Figure 1.1.3 symbolise the difference between this two new classification.

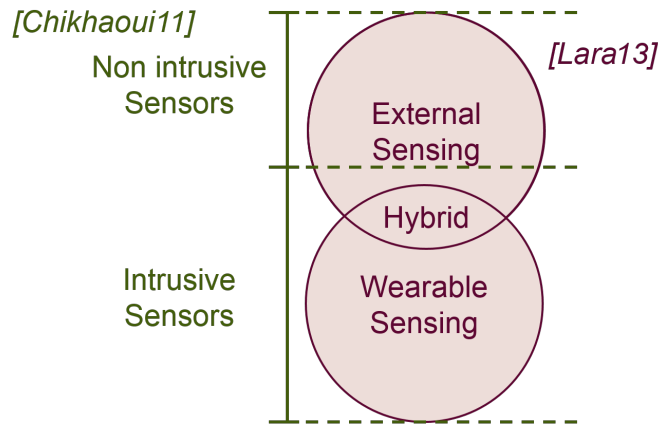


Figure 1.1.3: Sensors taxonomy

The acceptance issue is essential to allow a monitoring of inhabitants at home. And even if sensors generating semantically high information, like cameras, have an high performance in the field of activity of daily living monitoring, they are often considered too intrusive and raise problems of acceptance by monitored people ([Himmel et al., 2013](#)). This consideration also includes microphones.

Additionally, the efficiency of wearable sensors strongly depends on the ability and the willingness of the patients to wear them every day, and sometimes during the night. As in the case of cameras, this sensor technology also raises some problems of acceptance and, furthermore, is sometimes not compatible with the pathology of patients to be monitored (e.g. loss of memory).

To summarise, two issues involving human parameters have to be considered during the smart home instrumentation: the intrusiveness of the sensors and the ability of patients to live with. The following table shows the existing sensors, their location relatively to the human, their semantics levels and their compatibility with the two presented problems.

In this thesis, non-intrusive sensors compatible with any pathology are preferred to allow our method being applicable to the majority of cases. Therefore, in the proposed approaches, only binary environmental sensors are used, even if they only provide very low level of semantics information.

### 1.1.3 Activities of daily living definition and main topics

One of the possible ways to take care of persons health at home is to monitor their activities of daily living (ADL). ADL is defined by [Farlex \(2018\)](#) as following:

**Definition 1.5** (Activity of Daily Living ADL). *Tasks performed by people in a typical day that allow independent living. Basic activities of daily living (BADL) include*

Table 1.1.2: Review of existing sensors

Type of sensors	Location relatively to the human	Semantic level	Intrusiveness	Patient ability
Wearable vital sign sensors	Wearable sensing	High	Very intrusive	Sometimes incompatible
Cameras	External sensing	High	Very intrusive	Compatible
Other wearable sensors	Hybrid	Average	A little bit intrusive	Sometimes incompatible
Microphones	External sensing	Average	Intrusive	Compatible
Motion sensors	External sensing	Low	Non-intrusive	Compatible
Magnetic door sensors	External sensing	Low	Non-intrusive	Compatible
Vibration sensors	External sensing	Low	Non-intrusive	Compatible
Pressure sensors	External sensing	Low	Non-intrusive	Compatible
Flow sensors	External sensing	Low	Non-intrusive	Compatible
Temperature, light and humidity sensors	External sensing	Low	Non-intrusive	Compatible

*feeding, dressing, hygiene, and mobility. Instrumental activities of daily living (IADL) include more advanced skills such as managing personal finances, using transportation, telephoning, cooking, performing household chores, doing laundry, and shopping.*

*The ability to perform activities of daily living may be hampered by illness or accident resulting in physical or mental disability. Health care rehabilitation workers play a significant role in teaching people to maintain or relearn these skills so that they can achieve the highest possible degree of independence.*

Furthermore, in [Cook and Krishnan \(2015\)](#), authors precise than activities to monitor in a smart home include physical (or basic) activities as well as instrumented activities (introduced in [Lawton and Brody \(1969\)](#)). Therefore, the use of the generic term ADL is preferred.

In addition, as considered in [Chaaroui et al. \(2012\)](#), [Cook and Krishnan \(2015\)](#), [Ahad et al. \(2008\)](#) and [Fleury et al. \(2010\)](#), activities performed by a person can be decomposed into several actions. For instance “cooking” can be seen as the set of the actions “preparing pasta”, “preparing a ready-cook dish”, “ordering meal on the net”, etc. Moreover, actions can be described as a succession of elementary moves (or motions). This hierarchical decomposition of activities in actions and moves is represented in figure 1.1.4.

By reading the existing papers dealing with ADL, it appears four main topics based on the ADL studies: the activity discovery (AD) ([Kim et al., 2010](#); [Lara and Labrador, 2013](#)), the activity recognition (AR) ([Duong et al., 2005](#); [Lara and Labrador, 2013](#)), the activity prediction (AP) ([Mahmoud et al., 2013](#); [Krumm and Horvitz, 2006](#)) and the detection of behavioural deviation (DD) ([Chandola et al., 2012](#); [Forkan et al., 2015](#)). In this work, we focus on the AD and the AR.

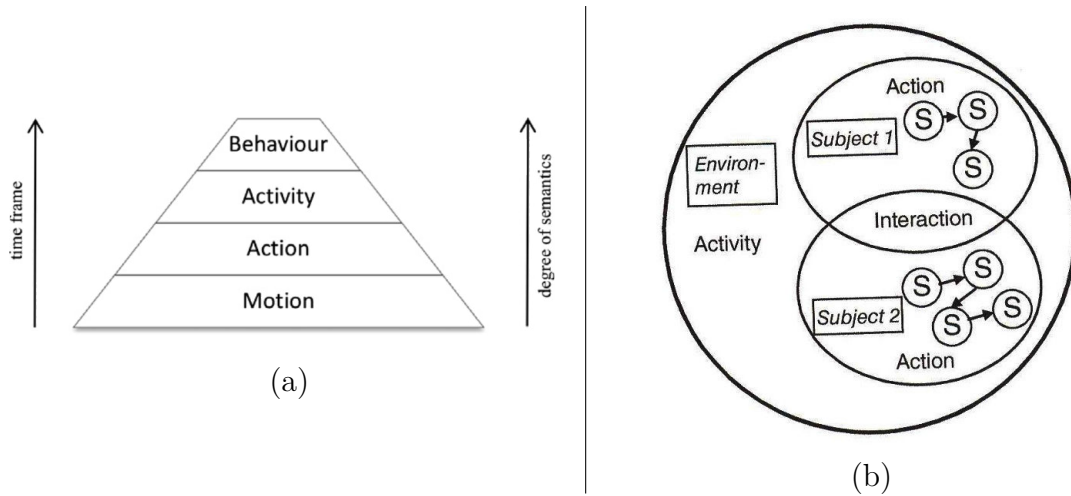


Figure 1.1.4: (a) Human behaviour analysis tasks - classification (Chaaroui et al., 2012) (b) Human behaviour degree of semantics (Cook and Krishnan, 2015).

#### 1.1.4 Activity discovery

First, ADL have to be modelled. When the modelled is generated by learning, we use the term of activity discovery defined as follows:

**Definition 1.6** (Activity Discovery AD). (Cook et al., 2013) *The activity discovery is an unsupervised learning algorithm to discover activities in raw sensor event sequence data.*

In the literature, a great variety of methods using different inputs and outputs can be found. A brief review of the major methods is now developed. In coherence with the sorting made in subsection 1.1.2, these methods are grouped by considering the semantic level of the sensors used. The semantic level of the generated models are also highlighted to improve the comprehension of the pros and cons linked to each method.

##### Inputs of high level of semantics

In Forkan et al. (2014), authors model human behaviour using expert knowledge and vital sign sensors. In addition, authors skip the ADL discovery process by choosing not to use data learning but rather ontological models as visible in table 1.1.3. Therefore, the human behaviour models of this method have a very high level of semantics since they correspond to very specific situations. With these models, it is possible to directly detect dangerous situation and quickly react in case of emergency. However, those models are fully constructed using expert knowledge, and therefore subject to human mistake and forgot.

In Duong et al. (2009), authors use cameras to detect the location of the inhabitant. Then, by using expert knowledge, authors link different successions of locations with activities to generate hidden semi-Markov models (HSMM) (Rabiner, 1989). A hidden Markov model (HMM) is a stochastic model of a process with an underlying part considered as non-observable. Furthermore, an HSMM is an HMM in which a duration knowledge is added. Figure 1.1.5 shows the dynamic Bayesian network graphical structure for HSMM with generic state duration distribution. At each time slice, a set of variables  $\mathcal{V}_t = \{x_t, m_t, y_t\}$  is maintained where  $x_t$  is the current state,  $m_t$  is duration variable of the current state, and  $y_t$  is the current observation. The duration  $m_t$  is a

counting-down variable, which not only specifies how long the current state will last, but also acts like a context influencing how the next time slice  $t + 1$  will be generated from the current time slice  $t$ .

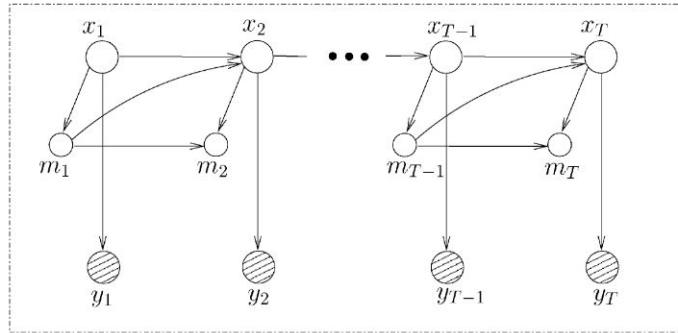


Figure 1.1.5: DBN representation for a standard HSMM. Shaded nodes represent observation. (Duong et al., 2009).

By performing known and wanted activities, authors use a Coxian distribution to efficiently model the duration information. This information added to the HSMM result in a novel form of stochastic model : the coxian hidden semi-Markov model (CxHSMM).

The discovery method presented uses expert-given HSMM as bases for the duration learning. According to the authors, the use of HMMs is suitable and efficient for learning simple sequential data. It is notable that, in this work, the information from cameras are quickly parsed to become a simple location information.

### Inputs of average level of semantics

In Lara and Labrador (2013), the AD, called *the training stage*, initially requires a time series dataset of measured attributes from individuals performing each activity. The time series are split into time windows to apply feature extraction and thereby filtering relevant information in the raw signals. Later, learning methods are used to generate an activity recognition model from the dataset of extracted features. Likewise, data are collected during a time window, which is used to extract features. Such feature set is evaluated in the priorly trained learning model, generating a predicted activity label (see figure 1.1.6).

A generic data acquisition is also an identified architecture for AD and AR systems, as shown in figure 1.1.7. In the first step, wearable sensors are attached to the person's body to measure attributes of interest such as motion, location, temperature, ECG, among others. These sensors should communicate with an integration device (ID), which can be a cellphone, a PDA, a laptop, or a customised embedded system.

The models thus obtained can be probabilistic or not. However, the need to split the data recorded during the learning period to "individuals performing of each activity" leads to record labels of the performed activity.

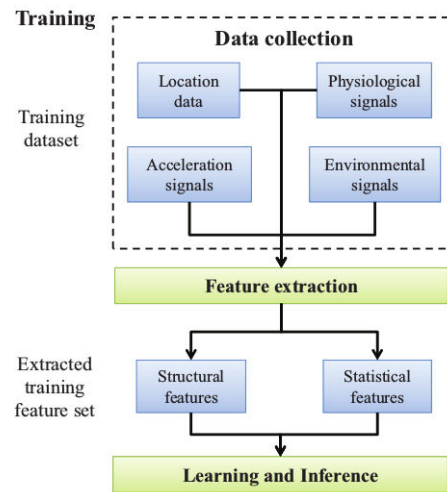


Figure 1.1.6: General data flow for training systems based on wearable sensors (Lara and Labrador, 2013).

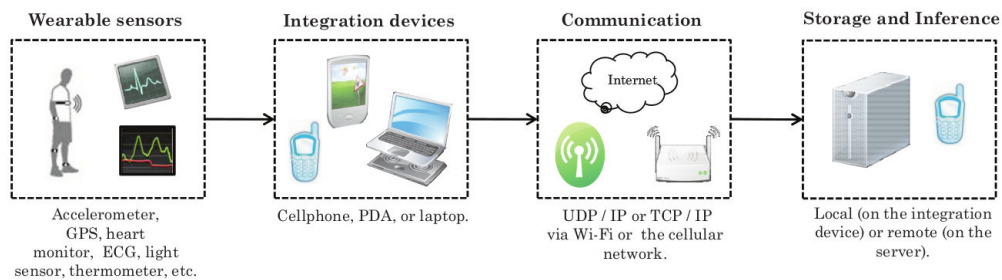


Figure 1.1.7: Generic data acquisition architecture for Human Activity Discovery and Recognition (Lara and Labrador, 2013).

### Inputs of low level of semantics

In Saives et al. (2015), authors propose a method to model, starting from a log of binary sensor events (rising and falling edges), the habits of the inhabitant. These models are extracted by sequence mining techniques and modelled by extended finite automata (EFA). The learned habits are then labelled by an expert. This AD method, base on the Agrawal and Srikant (1995) pattern mining method, is a black bock discovery method. However, as this pattern mining method distinguish each recurrent pattern, each aleatory and minor event inversion create a new pattern. The expert work is thus fastidious if treating data from a big smart home. As an output, authors gives a global map of activities represented by an EFA. This output model has a medium level of semantics since the main interesting part (i.e. the labelling) is done *a posteriori* by the expert.

In Cook and Krishnan (2015), authors presents several machine learning using binary sensors and individuals performing of each activity as inputs. Naïve Bayes classifier, Gaussian Mixture model, hidden Markov model, decision tree, support vector machine conditional random field are probabilistic models possible to generate with these inputs. The computed models have semantically high information since they are trained directly with adapted and labelled data.



The majority of the existing methods model ADL by probabilistic models. In addition to their natural robustness to small variations, those models have the advantage to be coherent with the human non-determinism. Therefore, in this thesis, a modelling of human ADLs by probabilistic model is preferred. However, when explained by the researchers, the generation of the models uses individuals performing of each activity to learn probabilities. This needed input signifies that, during the learning period, a log of the performed activity is recorded. Unfortunately, this information is in practice very difficult to obtain.

In [Tapia et al. \(2004\)](#), the monitored patient indicates which activity he is performing. Of course, the efficiency of this approach is confronted with the ability and the willingness of the person to declare his activity: in general, numerous reported activities errors are introduced in the database. In other works ([Gaglio et al., 2015](#)), experts are in charge of the enrichment of the database by studying sensor logs or by using cameras exclusively during the learning phase. This approach is expensive, intrusive and therefore risks changing the behaviour of the patient during the learning phase. In both cases, the labelling step is difficult and unreliable. That is the reason why, in the methods proposed in this thesis, the knowledge of actually performed activities during the learning phase is not required.

### 1.1.5 Activity recognition

Activity recognition methods are model-based approaches to monitor people. The used models can be given by an expert or obtained by learning (i.e. by applying an AD method). A definition of activity recognition is given in [Cook and Krishnan \(2015\)](#) as follows:

**Definition 1.7** (Activity Discovery AR). ([Cook and Krishnan, 2015](#)) *The field of activity recognition (AR) is concerned with the question of how to label activities from a sensor-based perception of the environment. The problem of AR is to map a sequence of sensor outputs onto a value from a set of predefined activity labels*

As for AD, a great variety of methods using different inputs and outputs can be found in the literature. In this subsection, existing methods will be classified according to the type of model used to model the activities.

#### **AR using descriptive model linked to data from vital signs sensors**

In [Forkan et al. \(2014\)](#), authors adopt the ontology-based context model ([Mocholi et al., 2010](#); [Devaraju and Hoh, 2008](#)) to recognise the performed activity or a dangerous situation. Figure 1.1.8 shows the proposed ontology-based context model based on OWL (web ontology language). Each context entity has attributes to describe some basic properties of the entity. Context entities are part of parent entity such as characteristics, diseases, preference, social and health ontologies are part of person ontology. Each of those entities has some more children to describe them. The relation among different entities is also shown.

The context space is described in four major entities:

- Person ontology is used to identify the user of the AAL system and his/her profile, diseases, health conditions, doctors, social interactions, and so on.
- Place ontology describes the current position of the user.
- Environment ontology is used to identify the conditions of surrounding environments. Environment has some impact on making decisions for assistive actions.

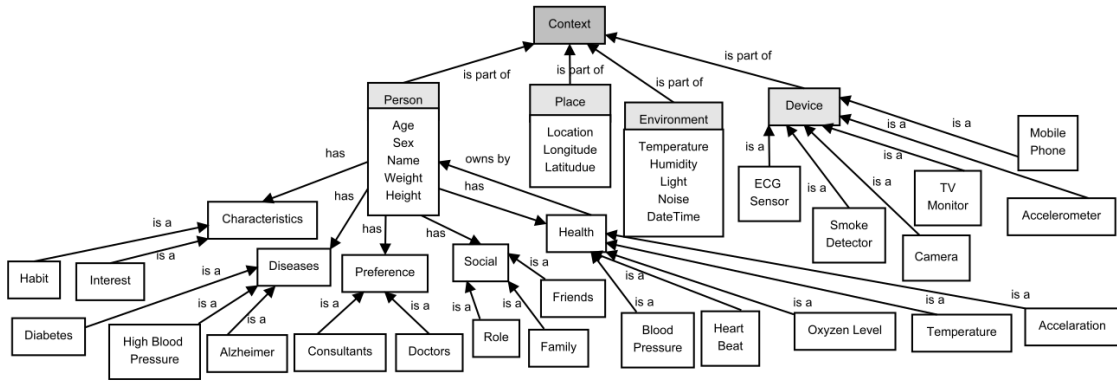


Figure 1.1.8: Context model using OWL (Forkan et al., 2014).

- Device ontology contains the details of the body sensors and devices of the system

To detect an activity or an abnormal situation, a service rule can be defined. For example, table 1.1.3 shows the service rules of detecting a possible heart attack by using the ontology model.

Table 1.1.3: Service rules for detecting a possible heart attack (Forkan et al., 2014).

Ontology	Instance	Raw data	Context attr.	Values
Person	User X	Profile	Age	≤ 65
Person	User X	Profile	Weight	≤ 80
Person	Disease	Profile	Cardiac patient	Have cardiac issue
Device	ECG sensor	ECG wave	Heart rate	Abnormal
Device	BP sensor	BP readings	Blood pressure	Normal or high
Device	PPG sensor	Sensor readings	O <sub>2</sub> consumption	Low
Device	Audio sensor	Sound wave	Breathing	Irregular
Device	Camera, radar, accelerometer	Video, images, 3D acceleration, motion path	Motion	Tripping or falling or flailing of arms or any rapid motion

By using these highly detailed models, authors can generate very precise information concerning the patient health state. Unfortunately, the given expert models are generic and not adapted to each human living each pathologies differently. As an example, the case of a heart attack for a human aging less than 65 years is not detected with the presented description table 1.1.3. Generating an adapted model for each human out of standards should be a fastidious and expensive process.

### AR using vision-based models

In Chaaraoui et al. (2012), authors make a great review on vision techniques applied to human behaviour analysis for AAL. According to the authors, it can be seen that at the motion, pose and gaze estimation level, several methods achieve robust and high success rates.

Mihailidis et al. (2004) are able to track the activity of hand washing to assist older adults with dementia. Multiple orders in the process can be correct, but not all of them. Their system is able to prompt the user if a necessary step is missing or the order of the implied actions is unacceptable. Vision is used as the only sensor in the developed system for two purposes: (1) tracking of hand location; and (2) tracking of step-specific object locations.

Related to this type of activity recognition, [Wu and Nevatia \(2005\)](#) stand out in activity recognition based on object use. These authors define activities as combinations of actions and objects and intend to recognise and track objects use in order to infer human activities. Object models are acquired automatically from video, whereas object identification is based on RFID labels. At the learning phase, the user wears a RFID bracelet which reads the RFID tags attached to the surrounding objects in a home environment. Assuming that the object being moved is always the object in use and that only one object is being moved at a time, the system learns the relationship between the segmented image and the active RFID tag using a dynamic Bayesian network. As arms and hands move with the objects, skin filtering is applied beforehand. At the test phase, the system works without the RFID data as objects are recognised by detecting SIFT features within the segmented area. These key points are matched based on maximum likelihood to the previously trained SIFT points.

In [Zhou et al. \(2008\)](#), activity recognition is approached differently. The individual silhouette is obtained at different positions of a living room. Grouped into 10–20 prototypes, each silhouette stores its centre, width and height and is manually labelled with a location. A fuzzy inference method is used to estimate the most likely physical location of test silhouettes. Location estimation and previously assigned coordinates enable average speed measurement, which is used besides location in order to recognise human indoor activities. A Hierarchical Action Decision Tree (HADT) is used to classify human actions using multiple levels. At the first level, human actions are classified based on location and speed. With K-means, clustering feature patterns are obtained; and activities of daily living, like walking or visiting the bathroom, are recognised.

All those presented methods are efficient in their field of appliances. Nevertheless, according to [Chaaraoui et al. \(2012\)](#), at higher levels, especially at behaviour, there is still a long way to go to achieve off-the-shelf products. Still, huge advances have been made in the last ten years. But the challenge to design and develop stable and general systems still persists, as most systems only solve specific problems in very particular environments.

### **AR using models linked to binary information**

In [Van Kasteren et al. \(2008\)](#) the authors describe all inhabitant activities by only one HMM. Then, authors recognise activities by applying the well-known Viterbi algorithm<sup>1</sup> ([Rabiner, 1989](#)). In fact, as explained in [Cook and Krishnan \(2015\)](#), inferring which sequence of labels best explains a new sequence of observations can be performed efficiently this algorithm. This dynamic programming technique is commonly used for HMM computations. If all activities are modelled in the same HMM, Viterby algorithm generates the most likely sequence of activity labels from a sequence of sensor observation. Unfortunately, the complexity of the model drastically increases with the number of activities and sensors. Furthermore, the used model has not intermediary semantic levels between activities and sensors and the precision of the recognition is not guaranteed.

In [Kellokumpu et al. \(2005\)](#), after converting video information to binary events traducing the human posture, the authors present a system that recognises a set of activity modelled by HMMs. Moreover, they classify activities by a probability that allows recognising the activity as being the one, which is represented by the most probable model.

---

<sup>1</sup>[https://en.wikipedia.org/wiki/Viterbi\\_algorithm](https://en.wikipedia.org/wiki/Viterbi_algorithm)

However, the previous methods can only compare models linked to the same sensor and event sets. On the contrary, in practice, activities are linked to different sensors because they are performed in different home areas and are realised by using diverse equipment in different spaces

In the majority of the presented methods, probabilistic models are used to estimate which activity is the most likely performed. For all the AR existing works, the recognition method is strongly linked with the models used to represent the activity. Therefore, if a new kind of models is used during the activity discovery, a new method, ideally based on existing ones, should be developed. The large use of probabilistic models comforts the decision to model activities by probabilistic models in this thesis.

### 1.1.6 Summary

Figure 1.1.9 summarises the context of this thesis: activity discovery and recognition are two principal missions of the assistance and monitoring of ADLs in health at home systems (HaH). HaH is the group of topics treating about the health monitoring of smart home inhabitants.

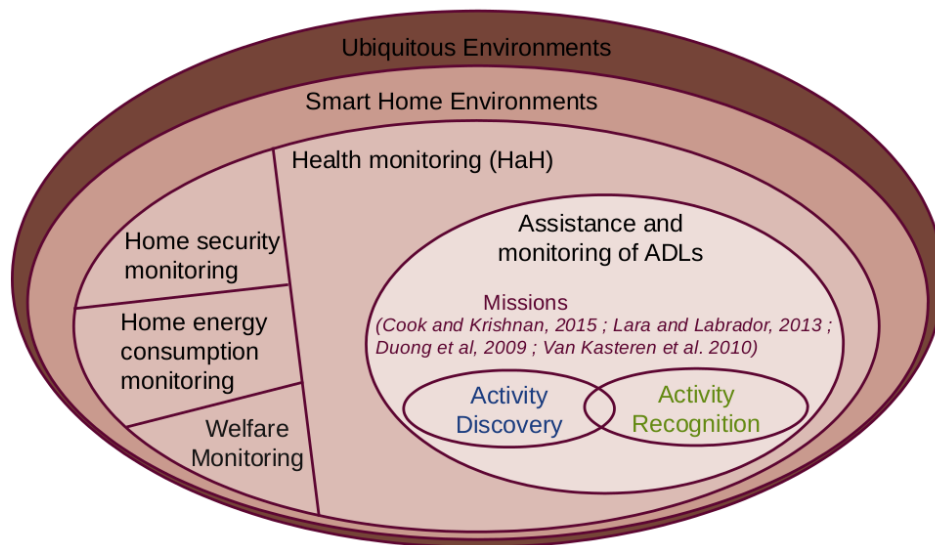


Figure 1.1.9: Thesis summarised context

Previously presented existing methods are listed in the figure 1.1.10. The mission performed in the paper, the level of semantics of the used sensors and their intrusiveness are shown.

In this thesis, the non-intrusiveness is preferred to the input knowledge semantic level. The problem statement of this thesis is presented in the next section 1.2.

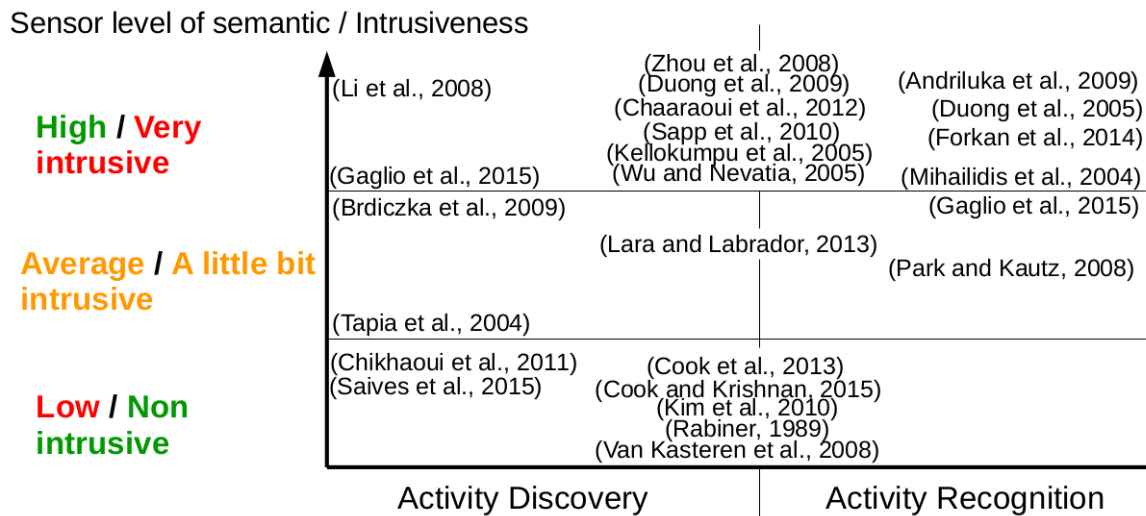


Figure 1.1.10: Existing methods, their sensor level of semantics and their intrusiveness.

## 1.2 Problem statement

Since the existing works are now presented, the problem statement of this thesis is developed in this section. The assumptions taken and their involvement are discussed and the proposed methods are summarised in a simple framework.

### 1.2.1 Global objectives

The literature review presented in section 1.1 highlights that managing the activities of daily living for dependent persons is a promising way to manage the increasing rate of the dependent population. Several methods more or less intrusive exist and are applicable following some conditions and assertions.

Even if the existing methods are different, some elements are common. The first one is the need to discover and recognise the ADLs of the inhabitants. Those two missions are the basic operations to perform in order to manage the human activities of daily living.

Therefore, the objective of this thesis is to develop a global framework to discover and recognise activities of daily living. As for the majority of the works in the literature, three sources of knowledge are authorised:

- *the medical staff* gives a list of ADL to be monitored corresponding to the inhabitant pathologies;
- *the sensors* of the smart home gives information during the inhabitant life. These sensor data can be recorded during a learning period or directly interpreted;
- *an expert* can, if needed, complete the basic information (flat floorplan, sensor locations,...) or those given by the two previously presented sources (*the medical staff* or *the sensors*).

In addition, in order to be applicable on a big scale and for the majority of the population, four points are considered, in this thesis, as unavoidable:

- The human nature has to be considered to choose the appropriate models;
- The patient privacy is a priority and its feeling about the intrusiveness of the used sensors have to be considered;

- Any choice has to be compatible with the patient pathology;
- The developed methods have to be applicable on a big scale and the information given by *the medical staff*, *the sensors* and *the expert* have to be viable and easy to obtain.

These points leads to make four assumptions described in the next section 1.2.2

### 1.2.2 Considered assumptions

In order to fulfil the objectives following the four unavoidable points presented just above, some practical and ethical decisions have to be taken and assumed. In this thesis, four main assumptions are made.

#### **Assumption 1: Activities are represented by probabilistic models**

The first unavoidable point is to consider the human nature to choose our model. It signifies that the human cannot be considered as a machine strictly repeating the same moves. Indeed, the human behaviour is, by nature, non-deterministic and may even be irrational. Therefore deterministic models and classical identification methods, more adapted to Cartesian and repetitive behaviour, cannot be used in the present problem.

During his life performance, a human can vary its way to perform an activity by inversing two moves during the "cooking" activity, for example. Indeed, as a medium variation, a human can choose, one day, to take the packet of pasta after boiling the water and the inverse another day. As a small variation, a human can open and close a useless cupboard to prepare tea because it forgot where is stored tea.

To be compatible with this human non-determinism, the chosen models have to be robust to variations. Therefore, in this thesis as in many paper in the literature, ADLs are modelled by using probabilistic models.

#### **Assumption 2: Only binary and environmental sensors are used**

The second unavoidable point is that the patient privacy is a priority and its feeling about the intrusiveness of the used sensors have to be considered. Therefore, as introduced before, cameras are rejected. Indeed, they can be considered as too intrusive and can be rejected by the patients (Himmel et al., 2013).

In addition, wearable sensors efficiency strongly depends on the ability and the willingness of the patients to wear them every day, and sometimes during the night. This sensor technology also raises problems of acceptance for the monitored inhabitant.

Furthermore, wearable sensors are sometimes not compatible with the pathology of patients to be monitored (e.g. loss of memory). This property is in contradiction with our third unavoidable point.

The elimination of the too intrusive and wearable sensors leads to the use of binary sensors only, such as motion detectors or door barriers. Such sensors are, furthermore, low cost. Unfortunately, by using only binary sensors, a difficulty due to the very low level of semantics information sensed exists.

#### **Assumption 3: The considered smart home is occupied by a single inhabitant**

According to Assumption 2, only binary sensors are used for both AD and AR. In case where several inhabitants are living in the same dwelling, it is not possible to distinguish which inhabitant is generating events observed through binary sensors. That is therefore necessary to make the assumption that a single-inhabitant is living in the smart home.

This assumption is quite restrictive but allows proposing a complete solution for AD and AR that is based on the use of binary sensors only. A way to relax this hypothesis is to consider that each inhabitant wears a sensor that allows identifying himself (e.g. a RFID sensor) and therefore knowing who has generated which event.

As we saw before, the use of this kind of sensors is incompatible with every pathology. However, in the case of multiple inhabitants with compatible pathology, the use of wearable sensors can be an acceptable trade-off between applicability and privacy.

The limit to a single inhabitant is assumed in this thesis but the presented method can be applicable to each person individually if a multiple inhabitant smart home is equipped with a RFID sensor that allows allocating each sensor occurrence to a unique person.

**Assumption 4: The knowledge of the actually performed activity is not required**

As precise before, several studies use the knowledge of performed activities during the learning phase to perform an efficient AD. Indeed, this information allows an easy decomposition of the observed data during the learning period and allows directly label the slit data. Unfortunately, as explained in chapter 1.1.4, this information is difficult to obtain and unreliable. Therefore, the use of this information is incompatible with the fourth unavoidable point since it is inapplicable and unreliable.

Therefore, in the presented thesis, the knowledge of the actually performed activity during the learning period is not required.

This Assumption 4 leads to a loss of information to compensate. The chosen solution is presented in the next section 1.2.3.

### 1.2.3 Proposed framework to discover and recognise ADLs

The objectives of this thesis is to propose new AD an AR methods following the four assumptions presented above. Figure 1.2.1 represent the proposed framework to perform this two objectives.

#### Activity Discovery

In the proposed framework, the AD is performed off-line to generate the output  $O_{AD}$  by using the inputs  $I_{AD}^1$  and  $I_{AD}^2$ .

In order to compensate for the loss of information implied by the Assumption 4, an additional expert knowledge has to be given. The retained solution is to use an expert knowledge based on the ADL definition. Therefore, with the goal to manage the observations, the expert has to give a hierarchical decomposition of the activities to be monitored into actions. Then, as sensors have a low level of semantics, their sensed information can be considered as having the same level of semantics as elementary moves. Therefore, expert details which sensor events are linked to which action. Those two hierarchical links *activities/actions* and *actions/sensors* are pertinent considering the definition 1.5 of ADLs. By doing this, the expert creates the input  $I_{AD}^1$  represented in figure 1.2.2. This input is called *hierarchical decomposition* in the rest of the manuscript.

The principle of the AD is to generate models using data observed during a learning period. Thus, the second input  $I_{AD}^2$  of the AD is a database obtained by recording the generated events during this observation. Due to Assumption 2, information recorded is binary. Then, according to Assumption 3, the learning database corresponds to the life of an inhabitant living alone in a smart home.

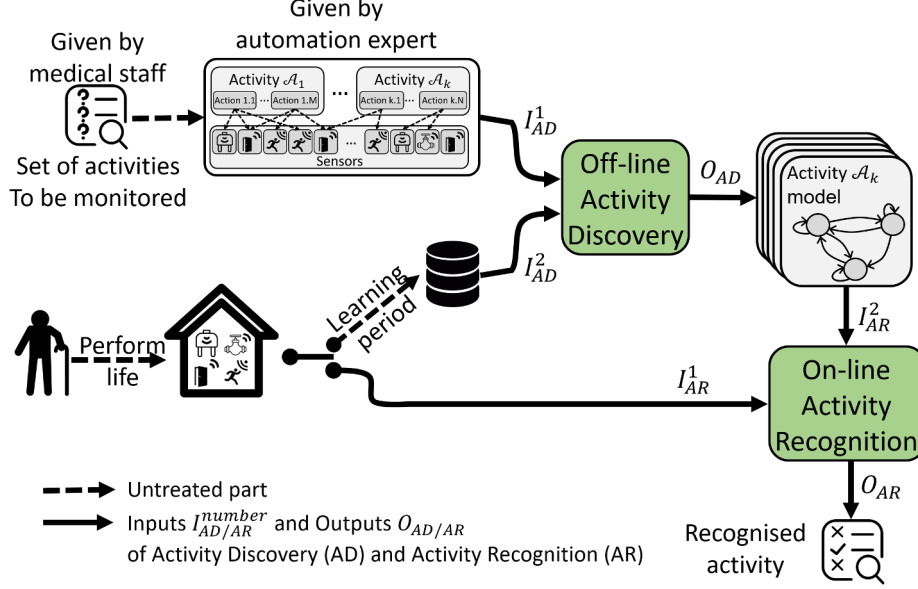
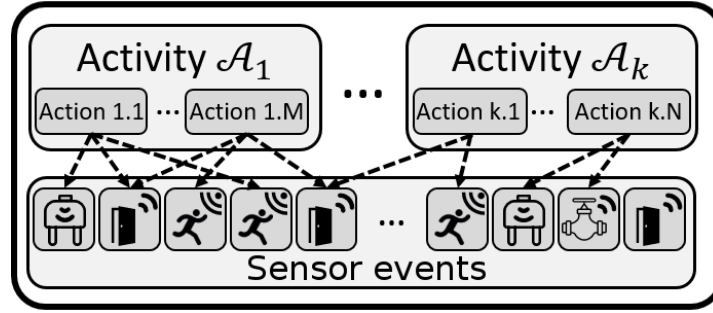


Figure 1.2.1: Activity discovery and recognition proposed framework.

Figure 1.2.2:  $I_{AD}^1$ : Hierarchical decomposition of activities into actions linked to sensor events given by an expert.

Starting from those two inputs  $I_{AD}^1$  and  $I_{AD}^2$ , the AR generates a set of probabilistic models ( $O_{AD}$ ) in accordance with Assumption 1, each one representing an activity (denoted  $\mathcal{A}_k$ ).

The used probabilistic model, its definition and notations and more generally the AD proposed method are given in chapter 3.

### Activity Recognition

The AR is a procedure performed on-line to recognise the activity  $O_{AR}$  actually performed by the inhabitant during its life by using the inputs  $I_{AR}^1$  and  $I_{AR}^2$ .

The AR is performed by using a direct observation of the events generated by the smart home. Thus,  $I_{AR}^1$  is the direct observation of this events. As explained in chapter 4,  $I_{AR}^1$  is not only composed by the last observed event, but by a log of the last observed events.

The second input  $I_{AR}^2$  is a set of models representing the activities to monitor. As the presented AR responds to a global framework containing AD and AR, the activity models are those obtained by applying our AD method. Therefore we have  $I_{AR}^2 = O_{AD}$ .



As output  $O_{AR}$  of the AR, the proposed AR protocol gives a probability, for each activity, to be performed. The section 4.6 explains how to conclude about the actually performed activity using those probabilities.

## Conclusion

In this chapter, basic definitions of ambient assisted living has been given and a global overview of the existing works in the field of activity of daily living discovery and recognition was presented. Smart home current technologies and activities models were presented. Assumptions for this thesis work was taken and explained to finally state the problem.

The taken assumptions are the following:

- Assumption 1 Activities are represented by probabilistic models;
- Assumption 2 Only binary and environmental sensors are used;
- Assumption 3 The considered smart home has a single inhabitant;
- Assumption 4 The knowledge of the actually performed activity is not required.

By considering these four assumptions, the problem can be summarised as follows:

*The objective of this thesis is to develop a global framework to discover and recognise activities of daily living of an inhabitant living alone in a smart home. This smart home has to be equipped with binary sensors only, expert labelling of activities should not be needed and activities can be represented by probabilistic models.*

# Chapter 2 Case Study

---

*"La scienza è il capitano, e la pratica sono i soldati."*  
- Leonardo da Vinci -

## Abstract

---

This chapter confronts three existing living labs, chosen among those that are open to an external usage or whose data are public, to the specific requirements of our works. Since none of them fulfil perfectly our demands, the smart flat we developed is described, as well as the experimental protocol used for tests.

---

## Contents

---

Introduction . . . . .	26
2.1 Existing living labs and public datasets . . . . .	26
2.1.1 Living lab of the Washington State University (WSU) . . . . .	26
2.1.2 Living lab of the Université de Sherbrooke (Domus) . . . . .	29
2.1.3 Living lab of the Massachusetts Institute of Technology . . . . .	31
Conclusion . . . . .	34
2.2 Ambient Assisted Living Test Area (AALTA) . . . . .	35
2.2.1 Description . . . . .	35
2.2.2 Activities to monitor . . . . .	38
2.2.3 Sensor placement . . . . .	39
2.2.4 Experimental protocol . . . . .	42
Conclusion . . . . .	44

---

## Introduction

In this thesis, applicative methods to monitor health at home system are developed. In order to illustrate and validate our propositions, a case study is needed. This case study should include a description of the used smart home, or living lab, and a well-detailed and fault-free database of activity performance. More precisely, the following criteria are mandatory.

Firstly, the equipped smart flat has to be perfectly known. It signifies that the floor-plan has to be given with details, the normal location of objects relevant considering the activities to be monitored (e.g. normal location of tea bags, coffee, pasta ...) has to be indicated and the exact location of sensors has to be supplied. This criterion is necessary to allow an expert hierarchical decomposition (see figure 1.2.2) sufficiently detailed. The knowledge of the object locations allows a fine decomposition. This expert knowledge is an essential input of the proposed activity discovery (AD) method and has to be reliable.

Secondly, since the activity recognition (AR) objective is to point out the performed activity out among several ones, it is important to have multiple activities to compare. Therefore, several activities have to be performed during the generation of the case study database.

Thirdly, the objective of this thesis is to discover and recognise complex activities using binary sensors with semantically poor information. Therefore, the case study has to be equipped with enough binary sensors to compensate for the low information by the number of them.

Fourthly, in accordance with the Assumption 3 enunciated in section 1.2.2, the data of the case study have to be generated by a person performing each activity alone and several times. Furthermore, since each activity model is linked to a way to perform it, the person performing the activities has to be the same every time.

Finally, in order to confront the result of our methods and the reality, the name of the performed activities and their realisation moments have to be recorded. As this knowledge is only needed to validate the methods, this need is not incompatible with our Assumption 4 rejecting this knowledge for real and non-experimental applications.

In this chapter, several existing databases are presented and their conformity with our criteria are discussed. Since no existing living-lab among the studied ones fulfil all the presented requirements, a new smart flat (AALTA) specially equipped for this thesis is presented.

### 2.1 Existing living labs and public datasets

In this section, three major living labs are presented. Each of them is well documented and their pros and cons are easily detectable. Unfortunately, due to the adopted point of view of our approaches, the documentation may not be exactly as wanted. The compatibility between the three presented living labs and our requirements are thus discussed.

#### 2.1.1 Living lab of the Washington State University (WSU)

As explained in [Cook et al. \(2009\)](#), to validate their algorithms, authors test them in a smart apartment testbed located on the Washington State University (WSU) campus. The CASAS testbed is equipped with motion and temperature sensors as well as analogue sensors that monitor water consumption and stove burner use (see figure

2.1.1). Voice over Internet Protocol (VOIP) captures phone usage and contact switch sensors are used to monitor usage of the phone book, a cooking pot, and the medicine container. Sensor data is captured using a customised sensor network and stored in an SQL database.

To provide physical training data, the research team brought 20 WSU undergraduate students to perform activities in the smart apartment. One at a time, student had to

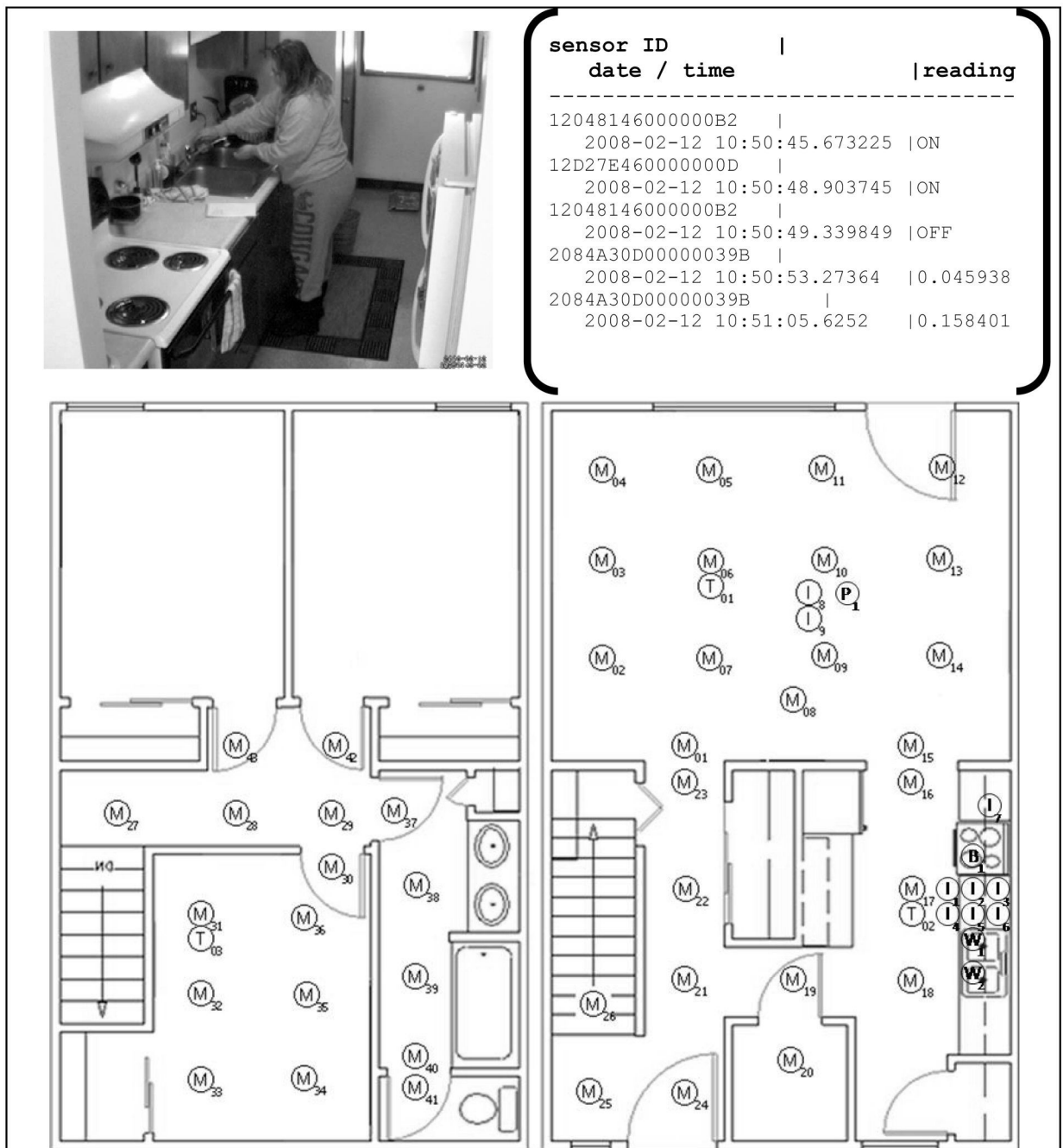


Figure 2.1.1: Resident performing “hand washing” activity (left). This activity triggers motion sensor ON/OFF events as well as water flow sensor values (right). Sensors in the apartment (bottom): monitor motion (M), temperature (T), water (W), burner (B), phone (P), and item use (I) (Cook et al., 2009).

perform the five following activities:

1. Telephone use: look up a specified number in a phone book, call the number, and write down the cooking directions given on the recorded message.
2. Hand washing: wash hands in the kitchen sink.
3. Meal preparation: cook oatmeal on the stove according to the recorded directions, adding brown sugar and raisins (from the kitchen cabinet) once done.
4. Eating and medication use: eat the oatmeal together with a glass of water and medicine (a piece of candy).
5. Cleaning: clean and put away the dishes and ingredients.

The selected activities include both basic and more complex ADLs that are found in clinical questionnaires (Reisberg et al., 2001). Noted difficulties in these areas can help to identify individuals who may have difficulty functioning independently at home (Schmitter-Edgecombe et al., 2009). As shown in figure 2.1.1, each sensor reading is tagged with the date and time of the event, the ID of the sensor that generated the event, and the sensor value.

The sensors used by Cook et al. (2009) and partially placed in figure 2.1.1 are:

- M01..M26: motion sensors;
- I01..I05: item sensors for oatmeal, raisins, brown sugar, bowl, measuring spoon;
- I06: medicine container sensor;
- I07: pot sensor;
- I08: phone book sensor;
- D01: cabinet sensor;
- AD1-A: water sensor;
- AD1-B: water sensor;
- AD1-C: burner sensor;
- asterisk: phone usage.

The data thus obtained are available at [ailab.eecs.wsu.edu/casas](http://ailab.eecs.wsu.edu/casas). It contains several text files named according to the participant and the task performed. Hence, the data recorded when the participant number  $i$  performed activity  $k$  is named " $p_i.tk$ ". In those files, sensors events are logged line by line. Each line represents the occurrence of one event as shown in figure 2.1.2. An event occurrence is thus characterised by its observance day (column 1) and time (column 2), the sensor name (column 3) and the event linked (column 4). In addition, a file "data" contains all of the sensor events in one file. In this new file, the performed activity is reported in a fifth column and the participant is not stored.

As the CASAS living lab was developed to test ADLs monitoring, it naturally fulfils some of our criteria presented in the introduction. That is the case with three of them:

- Several activities are considered during the data record (telephone use, hand washing, meal preparation, eating and medication use and cleaning), the second criterion is thus satisfied;
- Lots of binary sensors are implied in the living lab giving enough information to generate high-level information, the third criterion is thus filled;

2008-02-27	12:43:27.416392	M08	ON
2008-02-27	12:43:27.8481	M07	ON
2008-02-27	12:43:28.487061	M09	ON
2008-02-27	12:43:29.222889	M14	ON
2008-02-27	12:43:29.499828	M23	OFF
2008-02-27	12:43:30.159565	M01	OFF
2008-02-27	12:43:30.28561	M07	OFF
2008-02-27	12:43:31.491254	M13	ON
...			

Figure 2.1.2: Example of logs extracted from the CASAS file "p01.t1".

- The activities performed by the undergraduate student are well labelled, this fulfils the last criterion.

Unfortunately, two criteria are not fully fulfilled:

- Even if the activities are performed by a single inhabitant each time, they are not performed by a unique one. Indeed, multiple persons realise activities only once; this point discords with criterion four;
- Sensors placement is perfectly documented but the relevant object placements in the living lab is not precise enough to allow a detailed expert decomposition (first criterion).

### 2.1.2 Living lab of the Université de Sherbrooke (Domus)

In [Chikhaoui et al. \(2010\)](#), authors present another smart home: Domus. This smart home is a one-bedroom apartment, as shown in figure 2.1.3, mounted inside the University of Sherbrooke. It includes a bedroom, a bathroom, a kitchen, a dining room and a living room. During the experiments, six adults have performed the early morning routines which correspond to some basic activities of life. The experiments were held in the Domus smart home apartment in two steps.

First, users were asked to perform the early morning routine as they are supposed to do at home. This step was realised ten times per user.

Secondly, users were asked to repeat the same routine with an additional constraint: performing a given tea recipe which lasts at most 10 minutes. This step was realised five times per user.

In both series, the user can use any equipment available in the apartment, and decides the order of the activities that composed his routine. Each experiment lasted about 45 minutes.

During the experimentation, authors labelled the observed events in seven activities:

1. Wake up.
2. Use toilet.
3. Preparing breakfast.
4. Having breakfast.
5. Washing dishes.
6. Other activities (Unknown).
7. Preparing tea.

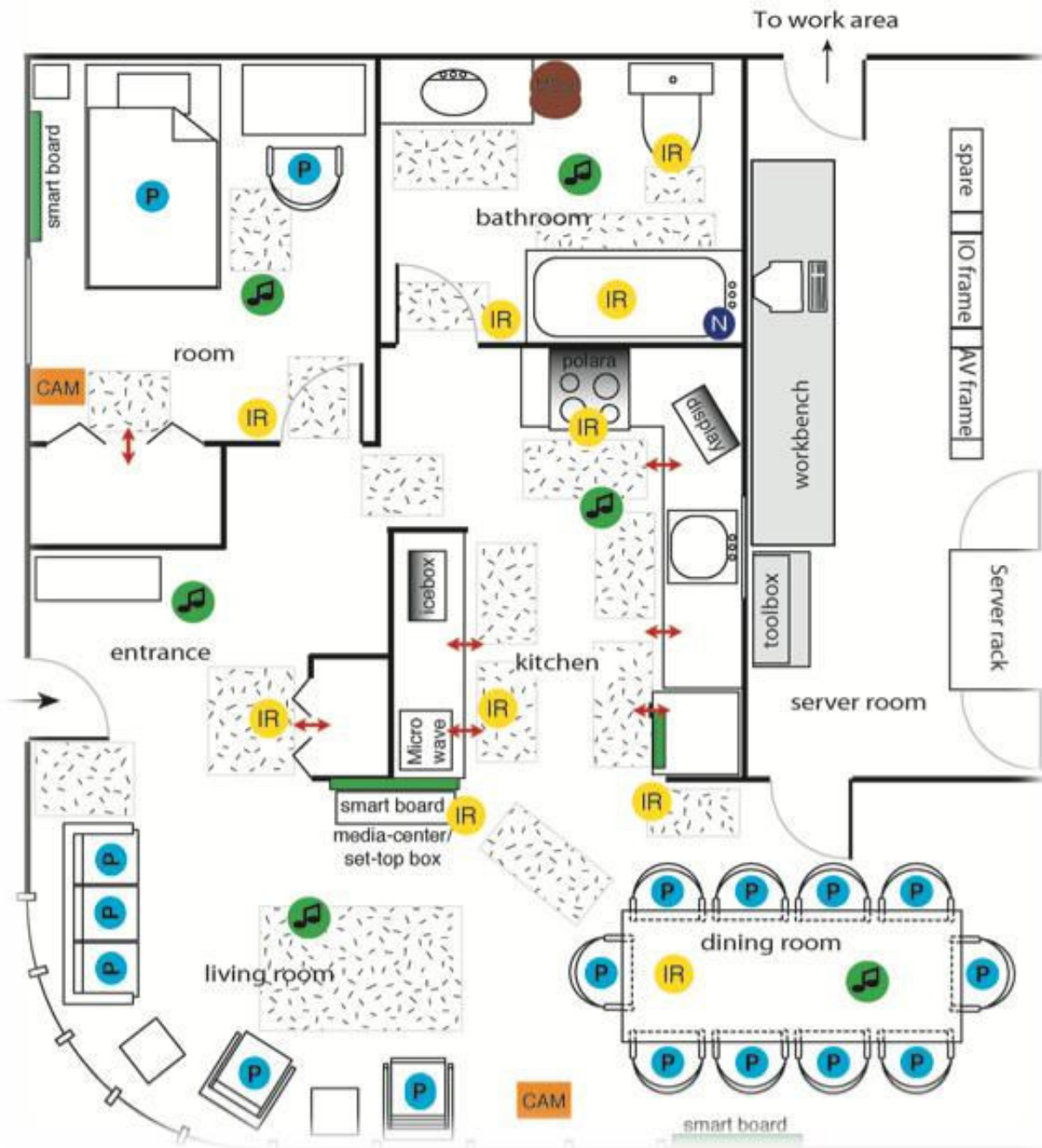


Figure 2.1.3: Domus smart home plan (Chikhaoui et al., 2010).

The Domus smart home is equipped with five different types of binary sensors:

- Infrared motion detectors;
- Pressure detector;
- Lamp light switches;
- Door contacts sensor;
- Flow sensors.

The obtained data are available at <https://www.usherbrooke.ca/domus/en/research/dataset/> and organised as follows. Each data from an experiment step  $i$  is classed in a

corresponding directory named "Domus Series  $i$ ". In each of these directories, there are six folders named according to the user number. Each user folder contains the sensor data files named per day "Day  $d.vna$ ". In these files, there is one sensor event on each line of the files presented as shown in figure 2.1.4. An event occurrence is characterised by its observance time (column 1), the sensor id (column 2), the sensor location in the smart home (column 3) and its new value (column 4). In addition, the folder contains also the activity annotation file that annotates the activities performed by the observed subjects during the experiments.

"9:8:18"	"5105"	"Lampe"	"Bedroom"	"Open"
"9:8:28"	"0215"	"Door"	"Bedroom"	"Open"
"9:8:34"	"5101"	"Lampe"	"Kitchen"	"Open"
"9:8:38"	"5105"	"Lampe"	"Bedroom"	"Close"
"9:8:42"	"0215"	"Door"	"Bedroom"	"Close"
"9:8:47"	"IR01"	"Infrared"	"KitchenSink"	"Close"
"9:8:50"	"F104"	"TapeColdWaterSink"	"Kitchen"	"Open"
"9:8:50"	"IR01"	"Infrared"	"KitchenSink"	"Open"
...				

Figure 2.1.4: Example of logs extracted from the file "Domus Series 1/User 1/Day 1.vna".

By analysing the Domus smart home dataset by regarding its conformity with our five criteria, it appears that four of them are respected.

- Several activities are considered (Wake up, Use toilet, Preparing breakfast, Having breakfast, Washing dishes, Unknown activities, Preparing tea). However, we can notice than only morning activities are tested. That validates the criterion two.
- An important number of events are implemented in the smart home and validates the third criterion.
- Activities are sufficiently performed by each subject to be compatible with the criterion four.
- The performed activities are labelled with start and end times. This fulfils criterion five.

Unfortunately, the sensor placement is not given with enough details. In addition, the position of the relevant tools in the smart home needed by the expert to construct a detailed hierarchical decomposition is not indicated. That does not fulfil the criterion one since it is impossible to viably construct an expert hierarchical decomposition.

### 2.1.3 Living lab of the Massachusetts Institute of Technology

As explained by the authors, in [Tapia et al. \(2004\)](#), two studies were run in two homes of people not affiliated with the MIT research group in order to collect data for developing and testing of the activity recognition algorithms. The first subject was a professional 30-year-old woman who spent free time at home, and the second was an 80-year-old woman who spent most of her time at home. Both subjects lived alone in one-bedroom apartment. 77 state-change sensors were installed in the first apartment and 84 in the second one. The sensors were left unattended, collecting data for 14 days in each apartment. During the study, the subjects used in the context-aware experience





Figure 2.1.5: (a) The top view of the apartments and the sensor distribution for subject one. (b) Sensor distribution for subject two. [Tapia et al. \(2004\)](#).

sampling method ([Intille et al., 2003](#)) to create a detailed record of their activities. Figure 2.1.5 shows the first apartment floor-plan. ([Tapia et al., 2004](#))

During the experiments, twenty-one activities were targeted and labelled. Those activities are decomposed into several actions not listed here.

- |                             |                             |                      |
|-----------------------------|-----------------------------|----------------------|
| 1. Employment work at home. | 8. Preparing a meal.        | 15. Home education.  |
| 2. Travel employment.       | 9. Preparing a snack.       | 16. Travel study.    |
| 3. Eating.                  | 10. Preparing a beverage.   | 17. Leisure.         |
| 4. Personal hygiene.        | 11. Meal Cleanup.           | 18. Travel social.   |
| 5. Sleeping.                | 12. Putting away groceries. | 19. Travel sport.    |
| 6. Talking on telephone.    | 13. Clean house.            | 20. Travel services. |
| 7. Resting.                 | 14. Outdoor chores.         | 21. Other.           |

For two weeks, the inhabitants lived normally at home and generated a database composed of 2323 and 1422 events. The sensors used are binary sensors labelled by:

- |                  |                    |                      |
|------------------|--------------------|----------------------|
| • Burner         | • Freezer          | • Shower faucet      |
| • Cabinet        | • Garbage disposal | • Sink faucet - cold |
| • Cereal         | • Jewelry box      | • Sink faucet - hot  |
| • Coffee machine | • Lamp             | • Stereo             |
| • Containers     | • Laundry Dryer    | • Telephone          |
| • Dishwasher     | • Light switch     | • Toaster            |
| • Door           | • Medicine cabinet | • Toilet Flush       |
| • Drawer         | • Microwave        | • TV                 |
| • DVD            | • Oven             | • Window             |
| • Exhaust Fan    | • Refrigerator     |                      |

The data thus obtained are available at <http://courses.media.mit.edu/2004fall/mas622j/04.projects/home/>. It contains two directories, one per subject. In the "text dataset", the data is stored in text format and comma separated values (.csv). Each directory contains three files:

- sensors.csv: containing information about the sensors;
- activities.csv: containing a description of the targeted activities and their decomposition in actions;
- activities\_data.csv: contains the observed events during the study.

The files "activities\_data.csv" are composed with blocks of five lines (see figure 2.1.6). In these blocks, the first line gives the performed activity and its date, begin time and ending time. The second one contains the ids of the sensors generating events in the occurrence time order. The third line corresponds to the list of the sensors generating events in the occurrence time order. The fourth line lists the rising edge times of the observed event. Finally, the fifth line gives the time corresponding to the falling edge of the observed event.

```

Preparing dinner,5/1/2003,16:15:7,17:40:7
115,115,79,73,73, ...
Microwave,Microwave,Cabinet,Cabinet,Cabinet, ...
16:14:24,16:17:10,16:17:45,16:17:50,16:18:9, ...
16:16:16,16:17:21,16:18:18,16:17:54,16:18:15, ...
Preparing lunch,5/1/2003,10:57:51,11:49:27
53,75,115,115,108, ...
Light switch,Light switch,Microwave,Microwave,Toaster, ...
11:3:13,11:3:17,11:5:39,11:7:17,11:8:10, ...
11:43:40,11:43:41,11:5:41,11:7:26,11:8:27, ...

```

Figure 2.1.6: Exemple of logs extracted from the [Tapia et al. \(2004\)](#) file "activities\_data.csv" of the "subject2" directory.

To facilitate the data treatment, the data can easily be parsed to appear with the same form as in the CASAS dataset 2.1.1. The data are thus as shown in figure 2.1.7.

05/01/2003	16:14:24	115	ON
05/01/2003	16:17:21	115	OFF
05/01/2003	16:17:45	79	ON
05/01/2003	16:18:15	73	OFF
05/01/2003	16:18:18	79	OFF
05/01/2003	16:18:28	89	OFF
05/01/2003	16:20:39	115	OFF
05/01/2003	16:21:00	84	OFF
...			

Figure 2.1.7: Example of readable presentation of [Tapia et al. \(2004\)](#) logs.

The presented living lab seems closed to be perfectly adapted to our work. Unfortunately, only three criteria are fulfilled:

- Several activities are considered during the data record (there are 21 activities in this case), the second criterion is thus satisfied;
- A huge number of binary sensors are implied in the normal life realisation of the two subjects, the third criterion is fulfilled;
- Activities are performed by a unique inhabitant for each smart home during a significant learning period. The fourth criterion is thus filled.

The following criteria are thus not respected:

- Even if the activities seem to be well-labelled, it appears by analysing them more precisely that the declarations made by the inhabitant are not already true. Since the labelling is performed directly by the inhabitant, the observed divergences can be due to human mistakes or decency considerations (voluntary or not). The last criterion is thus unfulfilled;
- Even if the decomposition of activities into actions is given, the specific object locations needed by the expert are not specified. Therefore, the links between actions and sensors cannot be done. The needed expert decomposition is thus impossible to obtain. It discords with the first criterion.

## Conclusion

By studying the datasets given by the existing databases, it appears that the five criteria needed to test and validate our methods are not fulfilled simultaneously in the existing works. Indeed, the necessity to generate an expert decomposition of activities into actions linked to events from the given data is limiting. This classification is feasible for an expert having access to the equipped home but is difficult to obtain in other cases. Therefore, in this thesis, it appears that we need to create our own living lab in order to know exactly the instrumentation and the performed activities. This perfect knowledge allows validating the methods using viable information.

The next section is thus dedicated to present the ambient assisted living test area (AALTA) developed at the ENS Paris-Saclay to test and validate our AD and AR methods.

## 2.2 Ambient Assisted Living Test Area (AALTA)

The smart flat we equipped for this Ph.D. is presented and described in this section. The activities to monitor we select to experiment our methods and sensors we installed are given and justified. Finally, the experimental protocol to obtain our test database is presented.

### 2.2.1 Description

Following our research of a smart home dataset corresponding to our needs, it appears that the best way to experience our methods is to develop our own living lab. Since the major need is a sufficient knowledge of the smart home to generate a dynamic expert decomposition as explained section 1.2.3, a large home is not necessary.

An existing flat in the ENS Paris-Saclay campus has been lent. The floor plan of this flat is given in figure 2.2.1. It can be decomposed into four different living zones:

- Zone A: entrance;
- Zone B: the bathroom;
- Zone C: the kitchen;
- Zone D: the sleeping zone.

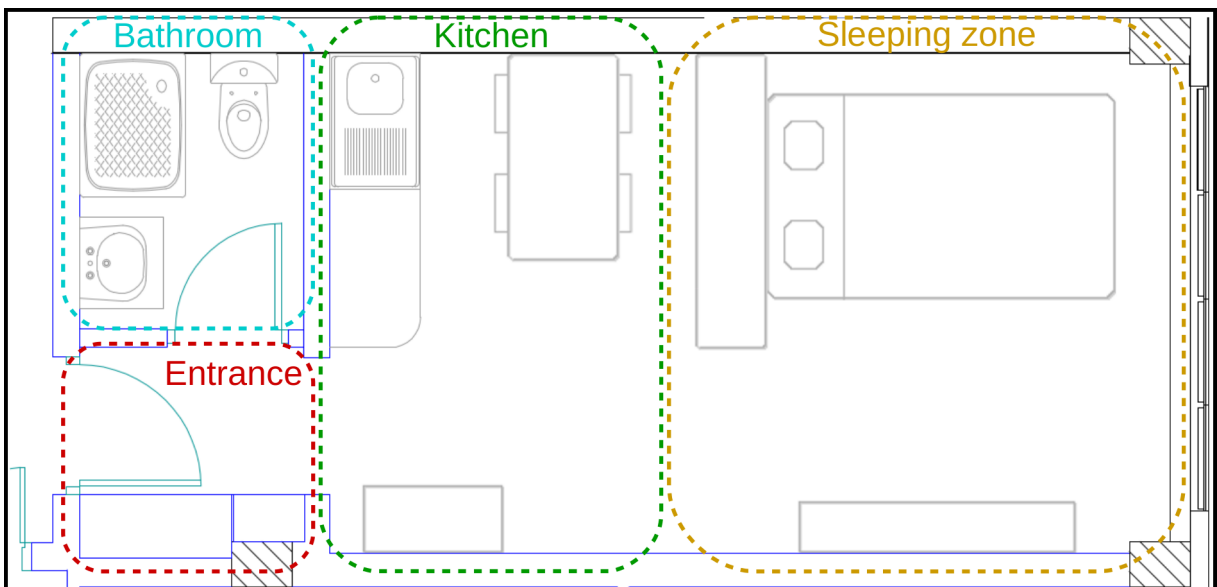


Figure 2.2.1: The smart flat with its zones

#### Zone A: entrance.

The entrance is a space open to the kitchen and the sleeping zone and near the bathroom. It is a transitory space between the kitchen and the bathroom. A wardrobe and a utility closet are installed in this entrance zone. Figure 2.2.2-a shows a picture of this zone.

#### Zone B: bathroom.

The bathroom is a simple room with a sink placed on a two-door cupboard, a shower cabin and toilets. Figure 2.2.2-b shows a picture of this bathroom.



Figure 2.2.2: Picture of the AALTA entrance (a) and bathroom (b).

**Zone C: kitchen.**

The kitchen was initially equipped with a fridge, five top cupboards, three bottom cupboards (one near the fridge and two sideboards near to the bed, at the limit of the kitchen zone), one top shelf, two hotplates a sink, five plugs, a table and four chairs. A picture of this zone is visible in figure 2.2.3.

Concerning the object placement, the sugar, the coffee and the coffee filters are placed on the first top cupboard starting from the extreme left (the closed with the entrance), called *Cupboard\_Left*. The tea bags are placed on the second one denoted *Cupboard\_CenterLeft*. The coffee and tea cups are placed on the third one noted *Cupboard\_CenterRight*.

**Zone D: sleeping zone.**

The sleeping zone is equipped with a bed, a window and a little shelf. This zone delimited by two of the three kitchen bottom cupboards is not separated from the kitchen using walls. Figure 2.2.4 shows this zone.



Figure 2.2.3: Picture of the kitchen (before equipping sensors and machines).



Figure 2.2.4: Picture of the sleeping zone.

### 2.2.2 Activities to monitor

As precise in section 1.2.1, in the real life application, the activities to be monitored are fixed by the medical staff and depend on the inhabitant pathologies. In this living lab, activities are chosen to be realistic while testing specific situations useful to discuss the obtained results. For example, the following four specific cases are interesting to observe:

1. An activity to monitor can be performed in a smart home zone in which no other activity is performed. In this case, whatever the sensor placement is, the observed events in this zone will obligatory be linked to this specific activity.
2. An activity can have two actions semantically closed but in practice very different to perform,
3. An activity can be so similar to another one that the distinction of them can be difficult;
4. In contrast to the previous point, two activities can be similar during a small part of their realisations but not during the rest of it.

Therefore, to be able to discuss results obtained for those cases, three activities to monitor have been chosen. These activities, denoted  $\mathcal{A}_1, \mathcal{A}_2$  and  $\mathcal{A}_3$  are selected among all activities encountered during the literature study. The chosen activities are thus:

- Activity  $\mathcal{A}_1$ : Cooking;
- Activity  $\mathcal{A}_2$ : Hot beverage preparation;
- Activity  $\mathcal{A}_3$ : Use bathroom.

with, as action composing those activities, the following:

- Action *prepare a ready-cooked dish* linked to activity  $\mathcal{A}_1$ ;
- Action *make pasta* linked to activity  $\mathcal{A}_1$ ;
- Action *make tea* linked to activity  $\mathcal{A}_2$ ;
- Action *make coffee* linked to activity  $\mathcal{A}_2$ ;
- Action *go to the toilets* linked to activity  $\mathcal{A}_3$ ;
- Action *have a shower* linked to activity  $\mathcal{A}_3$ .

Those hierarchical links are represented in the figure 2.2.5.

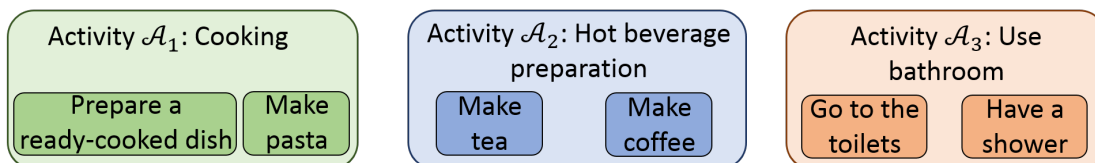


Figure 2.2.5: Hierarchical links between our activities and our actions.

As explained below, these four activities and their eight associated actions allow the four specific cases previously enumerated:

1. The activity  $\mathcal{A}_3$  is an activity performed in an allocated zone where no other activity is performed: the bathroom.
2. The actions *prepare a ready-cooked dish* and *make pasta* are linked to the same activity *cooking* and are semantically comparable since these are two ways to prepare food. But, in practice, the first action consists in taking a ready-cooked dish in the fridge to warm it using a microwave oven. In contrast, the second action implied to boil water, to cook pasta during some minutes, to drain them and potentially to add sauce. These two actions are thus in practice radically different.
3. The activity  $\mathcal{A}_2$  is nearly identical to the *prepare pasta* action of the activity  $\mathcal{A}_1$  when the inhabitant prepares tea by making water to boil on the hotplates.
4. Finally, by preparing the coffee, the activity  $\mathcal{A}_2$  shares some action with the action *prepare pasta* of  $\mathcal{A}_1$  since it is necessary to fill the machines with water.

### 2.2.3 Sensor placement

Since the smart flat and the activities to monitor are now described, it is possible to choose an efficient implementation of sensors to discover and recognise them.

In order to well understand the output given by the binary sensors, let's recall what represents an event for a binary sensor. Each sensor, denoted by an explicit name as *Kitchen|Cupboard\_Bottom|Open* can generate two events: one linked to the rising edge of its binary information (thus denoted *sensorName|1*), the other one linked to the falling edge (*sensorName|0*) (see figure 2.2.6).

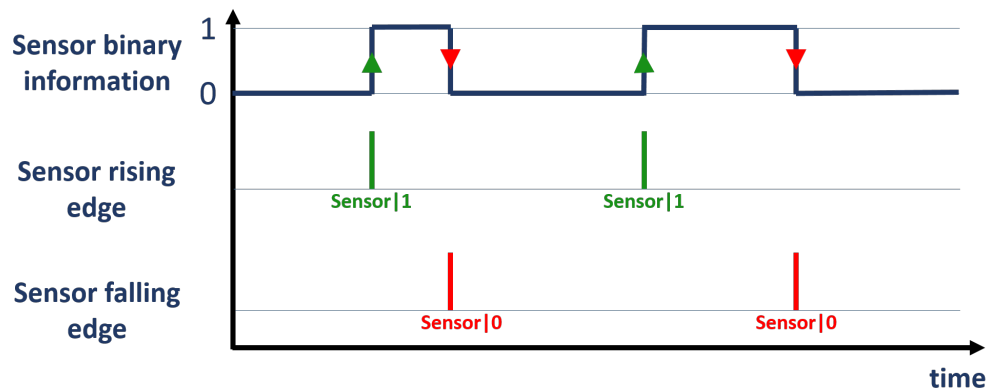


Figure 2.2.6: Event emission from sensor binary information.

This flat was instrumented to detect the presented activities using four kinds of binary sensors shown in figure 2.2.7.

- Door sensors, picture (a);
- Coupled door sensors and motion detectors, picture (b);
- Water flow sensors, picture (c);
- Smart outlets, picture (d).





Figure 2.2.7: Installed kinds of binary sensors.

Table 2.2.1: List of events logged in AALTA.

Id	Event name	Id	Event name
$e_1$	<i>Bathroom Door_PIR Open 0</i>	$e_{22}$	<i>Kitchen Fridge_PIR Open 1</i>
$e_2$	<i>Bathroom Door_PIR Open 1</i>	$e_{23}$	<i>Kitchen Fridge_PIR Presence 1</i>
$e_3$	<i>Bathroom Door_PIR Presence 1</i>	$e_{24}$	<i>Kitchen Hotplate_bottom Power 0</i>
$e_4$	<i>Bathroom Shower_Door Open 0</i>	$e_{25}$	<i>Kitchen Hotplate_bottom Power 1</i>
$e_5$	<i>Bathroom Shower_Door Open 1</i>	$e_{26}$	<i>Kitchen Hotplate_top Power 0</i>
$e_6$	<i>Bathroom Shower_Water flow Flow 0</i>	$e_{27}$	<i>Kitchen Hotplate_top Power 1</i>
$e_7$	<i>Bathroom Shower_Water flow Flow 1</i>	$e_{28}$	<i>Kitchen Kettle Power 0</i>
$e_8$	<i>Bathroom Sink_Water flow Flow 0</i>	$e_{29}$	<i>Kitchen Kettle Power 1</i>
$e_9$	<i>Bathroom Sink_Water flow Flow 1</i>	$e_{30}$	<i>Kitchen Microwave_oven Power 0</i>
$e_{10}$	<i>Bathroom Toilets_Water flow Flow 1</i>	$e_{31}$	<i>Kitchen Microwave_oven Power 1</i>
$e_{11}$	<i>Kitchen Coffee_Machine Power 0</i>	$e_{32}$	<i>Kitchen Sideboard_Left Open 0</i>
$e_{12}$	<i>Kitchen Coffee_Machine Power 1</i>	$e_{33}$	<i>Kitchen Sideboard_Left Open 1</i>
$e_{13}$	<i>Kitchen Cupboard_Bottom Open 0</i>	$e_{34}$	<i>Kitchen Sideboard_Right Open 0</i>
$e_{14}$	<i>Kitchen Cupboard_Bottom Open 1</i>	$e_{35}$	<i>Kitchen Sideboard_Right Open 1</i>
$e_{15}$	<i>Kitchen Cupboard_CenterLeft Open 0</i>	$e_{36}$	<i>Kitchen Sink_Water flow Flow 0</i>
$e_{16}$	<i>Kitchen Cupboard_CenterLeft Open 1</i>	$e_{37}$	<i>Kitchen Sink_Water flow Flow 1</i>
$e_{17}$	<i>Kitchen Cupboard_CenterRight Open 0</i>	$e_{38}$	<i>Kitchen Wardrobe Open 0</i>
$e_{18}$	<i>Kitchen Cupboard_CenterRight Open 1</i>	$e_{39}$	<i>Kitchen Wardrobe Open 1</i>
$e_{19}$	<i>Kitchen Cupboard_Left Open 0</i>	$e_{40}$	<i>Entrance Door_PIR Open 0</i>
$e_{20}$	<i>Kitchen Cupboard_Left Open 1</i>	$e_{41}$	<i>Entrance Door_PIR Open 1</i>
$e_{21}$	<i>Kitchen Fridge_PIR Open 0</i>	$e_{42}$	<i>Entrance Door_PIR Presence 1</i>

More specifically, the list of the forty-two sensor events used in AALTA is given in Table 2.2.1.

The twenty sensors generating the forty-two events are placed on the smart home as represented in the floor-plan figure 2.2.9 and in the picture visible figure 2.2.8.

Starting from these activities to monitor and the detailed knowledge of the equipped flat, it is possible to give a coherent expert decomposition of activities in actions linked to events. The decomposition we retained is detailed in figure 2.2.10. This decomposition is used in chapter 3 to discover the activities to monitor.

By analysing this decomposition, we can see that, as wanted, the four different cases of more or less important overlapping of sensors between activities are well present. We

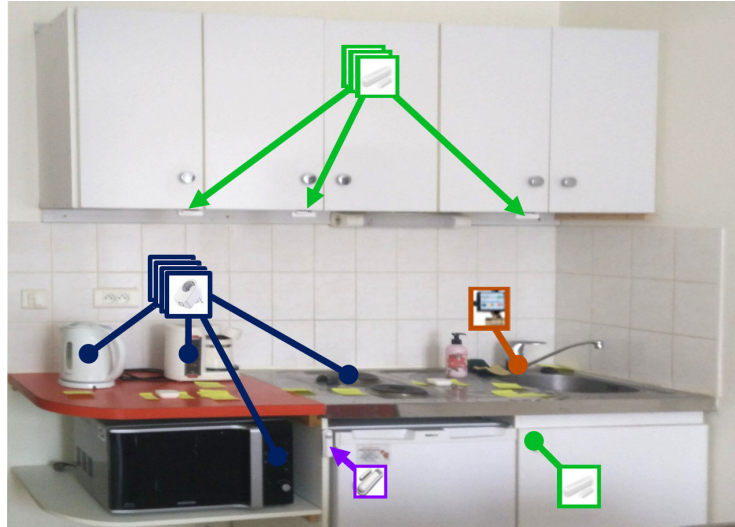


Figure 2.2.8: Kitchen view of the smart flat.

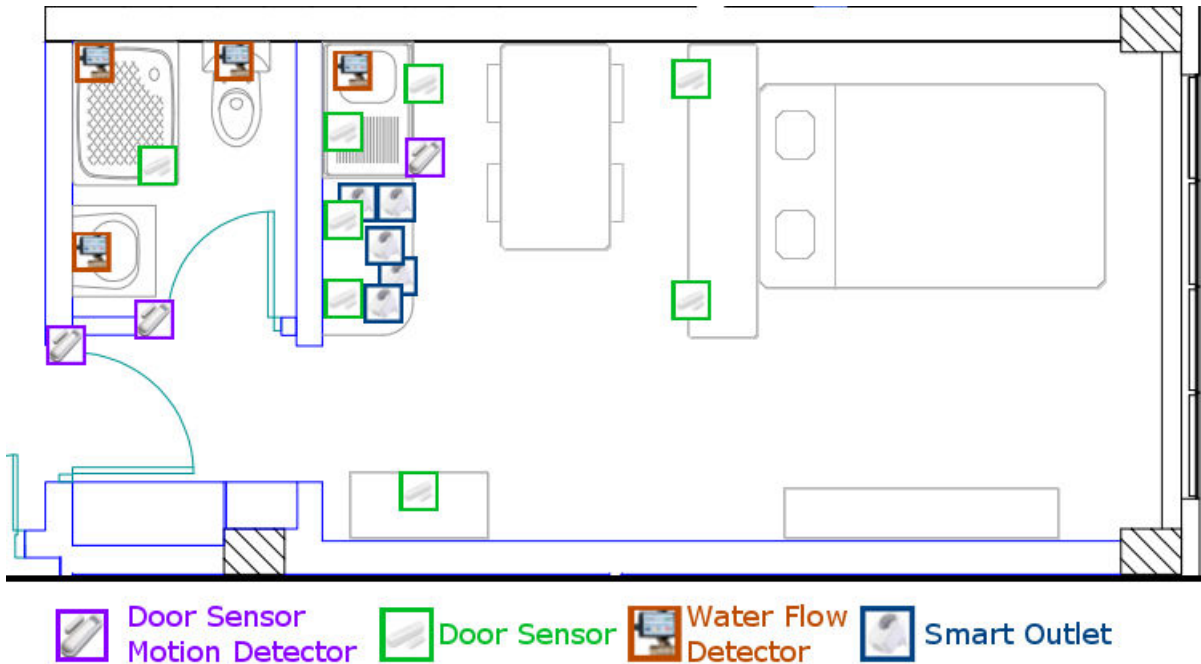


Figure 2.2.9: The smart flat with its binary sensors

have activities with actions sharing some sensors, others sharing a lot of sensors and the activity  $\mathcal{A}_3$  totally isolated.

The figure 2.2.11 briefly summarises our living lab network. Our sensors use the Z-wave wireless communication protocol to emit or receive instructions. The Z-wave protocol is a mesh network using low-energy radio waves to communicate from appliances to appliances. And USB Z-wave controller plugs on a computer manage the networks and receipt all information. This information is then transmitted to the computer to analyse. The open-source software Jeedom, installed on the computer, is in charge to record these information and to facilitate the human-machine interaction. This software

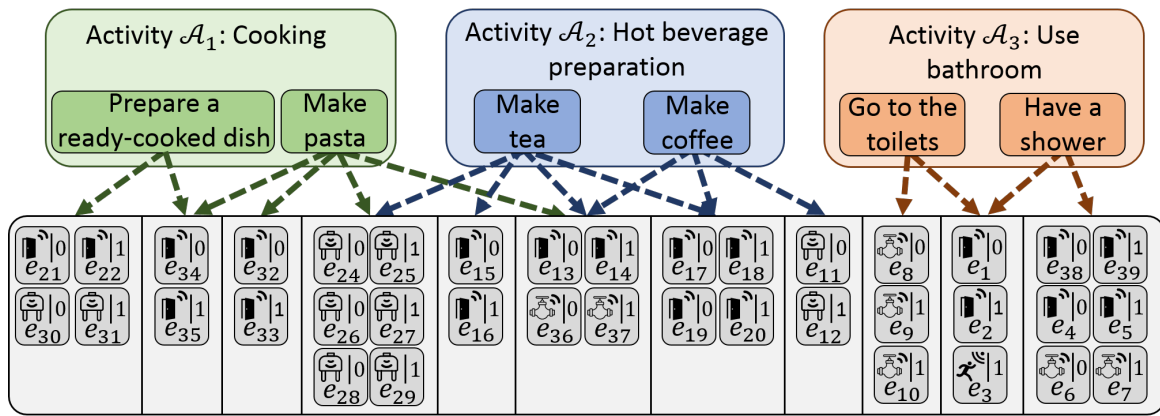


Figure 2.2.10: Expert hierarchical decomposition of activities to monitor

is available at <https://www.jeedom.com/site/en/soft.html>.

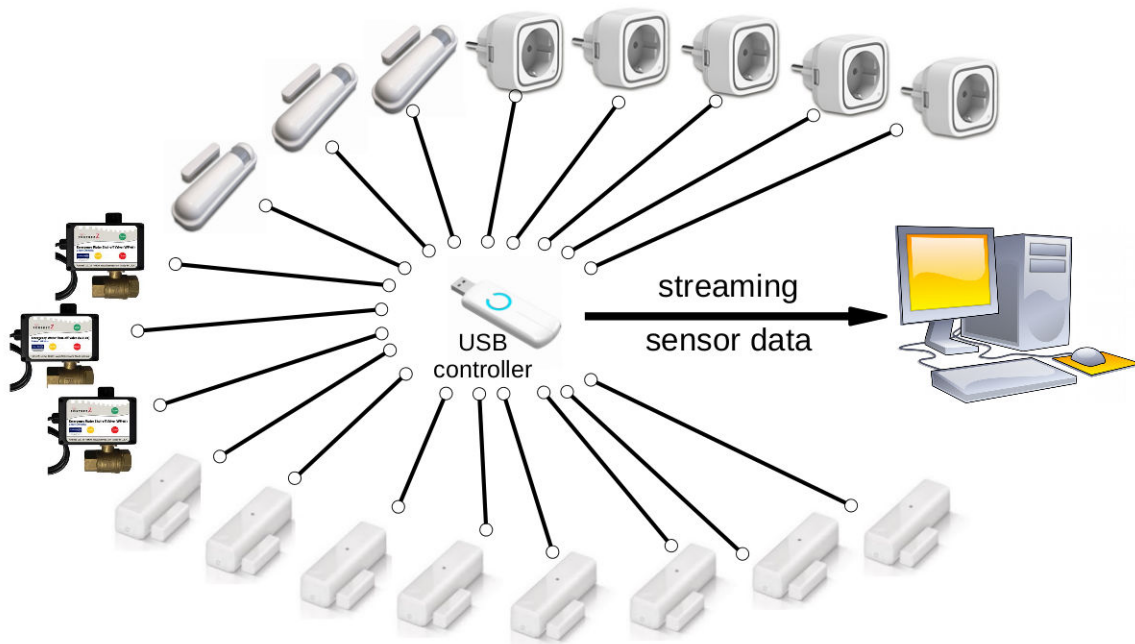


Figure 2.2.11: Living lab sensor networks

Finally, the data generated by our instrumentation during experiments have the format presented in figure 2.2.12.

Since the instrumented smart flat and the expert hierarchical decomposition of activities are perfectly known, we can define an experimental protocol to generate and use a test database.

### 2.2.4 Experimental protocol

In order to estimate the robustness of the approach, during our experiment phase, activities  $\mathcal{A}_1$ ,  $\mathcal{A}_2$  and  $\mathcal{A}_3$  are realised and observed a huge number of times by introducing the following variations:

2017-04-04	10:11:09	$e_{22}$
2017-04-04	10:11:16	$e_{21}$
2017-04-04	10:11:21	$e_{31}$
2017-04-04	10:11:32	$e_{23}$
2017-04-04	10:11:32	$e_{30}$
2017-04-04	10:11:43	$e_{35}$
2017-04-04	10:12:01	$e_{34}$
...		

Figure 2.2.12: Example of AALTA logs.

- the insertion of noisy events (i.e., events not linked with the performed activity) during their realisation, for instance by wandering in the flat;
- some actions are interrupted;
- the execution order of elementary moves composing actions is changed;
- the action *make tea* is realised by two different ways: using the kettle or boiling water with hotplates.

These variations focus the majority of possible noisy phenomena occurring during a normal human life.

These activity realisations, split in different parts, compose a library of modular activity performance. By ordering them in different order and following different rules, it is possible to test several particular situations. On the one hand, to realise simple test, we can use activities performed without noise timely place in an optimal order. On the other hand, to test more difficult situations such as interrupted activities or a direct succession of activities, we can use all kind of activity performance timely closed or broken.

In order to run our methods onto an acceptable dataset, a learning sequence was creating using a library containing realisations of activities to monitor. More precisely, this database is generated using recorded activity instances placed in a random time order and separated by a random number of random noisy events not belonging to the performed activities. The resulting sequence is composed of 2087 events corresponding to twenty realisations of each activity. This database generation method is presented in figure 2.2.13.

First, to validate our methods, the activity recognition will be run over the learning database to verify that we are able to recognise the activities used during the discovering phase. Then, additional tests are run to analyse the robustness of our methods to successions of activities. For all those experiments, the details about the placed activities and their positions in the constructed database are kept. This information is used to compare the methods results to the reality. This knowledge is for the validation process only and is not needed for an application in the real world.

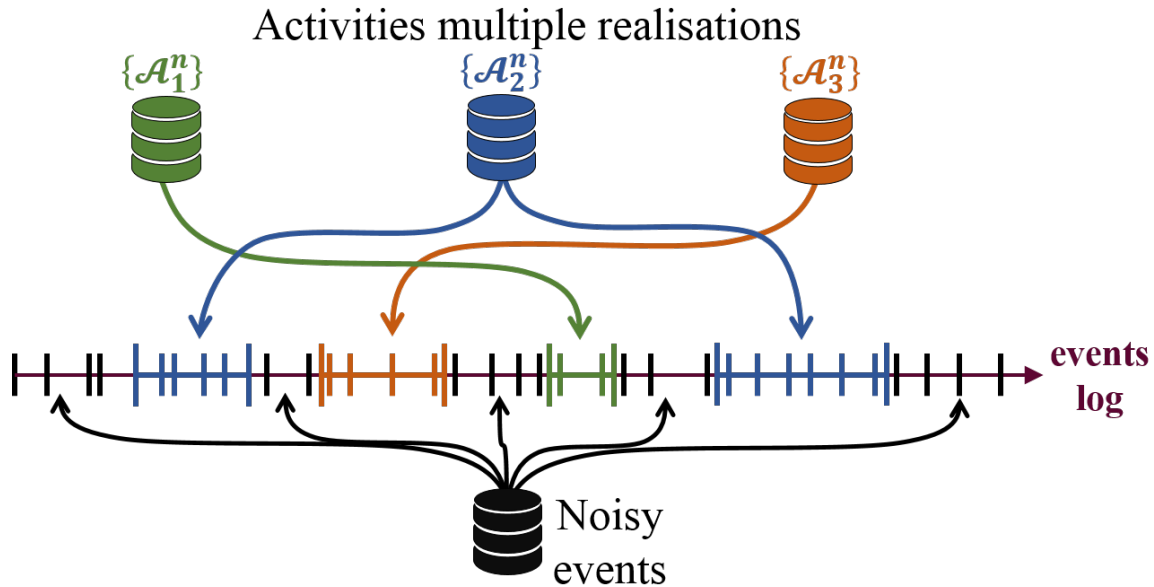


Figure 2.2.13: Structure of the test sequence of observed events

## Conclusion

Since the topic treated in this thesis is directly linked to an actual societal problem needing concrete solutions, all methods presented in this thesis have to be applicable on a real case. The methods development cannot be decoupled with the technical application. Therefore, in this chapter, several living labs were presented and their usability to apply our methods using their datasets was discussed. Unfortunately, the restrictions linked to the assumptions made in chapter 1 make the currently known living labs not fully adapted to our methods. These incompatibilities are mainly due to a lack of information concerning the equipped smart homes.

To deal with this problem, a new living lab was developed at the ENS Paris-Saclay. This ambient assisted living test area (AALTA) is presented in section 2.2. Object and sensor placements are given and the expert decomposition needed in the activity discovery method (presented in chapter 3 of the manuscript) is detailed. Finally, the experimental protocol used to generate out test databases is developed and explained.

# Chapter 3

## Activity Discovery

---

*"If you want to know someone, don't listen to what he says, but look what he does."*  
- Tenzin Gyatso, His holiness the 14th Dalai Lama-

### Abstract

---

This chapter presents a probabilistic approach allowing to model activities of daily living under the form of probabilistic finite-state automata. This method uses as input a log of events recorded during a learning phase and a hierarchical decomposition of activities to monitor to actions linked to the instrumentation of the dwelling. Finally, this approach is applied to the experimental smart flat to illustrate and validate its discovery method.

---

### Contents

---

Introduction . . . . .	46
3.1 Models and notations . . . . .	46
3.2 A systematic procedure for models generation . . . . .	49
3.2.1 Generation of PFA structure . . . . .	49
3.2.2 Database of event logs exploration . . . . .	53
3.2.3 Probabilities computation . . . . .	55
3.3 Application to the Case Study . . . . .	61
3.3.1 Generation of PFA structure . . . . .	61
3.3.2 Database of event logs exploration . . . . .	63
3.3.3 Probabilities computation . . . . .	67
3.4 Discussion . . . . .	73
Conclusion . . . . .	73

---

## Introduction

In this chapter, we present the first contribution of this thesis: a new activity discovery (AD) method. We need to develop this innovative approach because of our Assumption 4 incompatible with existing methods. Indeed, Assumption 4, stated in section 1.2.2, rejects the possibility of labelling activities actually performed by the monitored person during the learning period. Unfortunately, this information is usually one of the mandatory inputs of AD methods.

Moreover, Assumption 2 consists in treating information coming from environmental binary sensors only. Therefore, the method is free from performed activity knowledge and uses only environmental and binary sensors.

One major asset of this approach is its portability. Indeed, it is applicable to each smart home regardless of the inhabitant pathologies (e.g. a patient suffering from Alzheimer do not need to help the process).

This approach models each activity to monitor by probabilistic finite-state automaton (PFA). Lost information due to the rejection of the performed activity knowledge is compensated by the addition of a specific expert knowledge which gives a hierarchical decomposition of the activities into a set of actions observable using a given subset of events among the smart home ones.

First of all, a theoretical picture of this new method is presented with the definitions and notations that are needed for a better understanding.

Then this concept is illustrated with the study of a specific case that has been described in the previous chapter.

### 3.1 Models and notations

In order to develop the proposed method, we first need to present the general notions of discrete event system modelling such as alphabets and probabilistic finite-state automata (PFA), as well as specific notations for this method.

As a beginning, we provide definitions for sensor events and activity instance as defined in [Cook and Krishnan \(2015\)](#).

**Definition 3.1** (Sensor event). *An activity discovery method receives data from sensors that are used to perceive the state of an individual and/or an environment. One of the input data of an AD method is a sequence of sensor events. Each sensor event  $e_i$  can be seen as a tuple  $e_i = \langle s, m \rangle$  where  $s$  denotes a sensor ID and  $m$  denotes the sensor message. As we threat only binary sensors (Assumption 2), the message can take two values (rising and falling edge) as explain in figure 2.2.6.*

**Definition 3.2** (Activity instance). *We define an activity instance or activity occurrence as a subsequence of  $n$  sensor events  $e_1 e_2 \dots e_n$ . An activity represents the collection of all of its instances. An activity model may represent an activity as an abstraction of this collection of activity instances.*

First of all, we give the formal definition of the expert decomposition as following:

**Definition 3.3** (Expert decomposition). *The expert decomposition needed in this thesis can be described as a tuple:*

*decomposition =  $\langle lActivities, lActions, \Sigma_{Tot}, Link_{Activity/Action}, Link_{Action/Event} \rangle$ ,*  
*where:*

- $lActivities$  is the list of activities to be monitored  $\mathcal{A}_k$ ;
- $lActions$  is the list of actions  $Action\ k.h$  composing the activities;
- $\Sigma_{Tot}$  is the list of events  $e$  generated by the sensors in the smart home;
- $Link^{kActivity/Action} : lActivities \rightarrow lActions^*$  is a function linking the activities and the actions;
- $Link^{kAction/Event} : lActions \rightarrow \Sigma_{Tot}^*$  is a function linking the actions and the events.

As seen in chapters 1 and 2, this expert decomposition can be graphically represented as in figure 3.1.1.

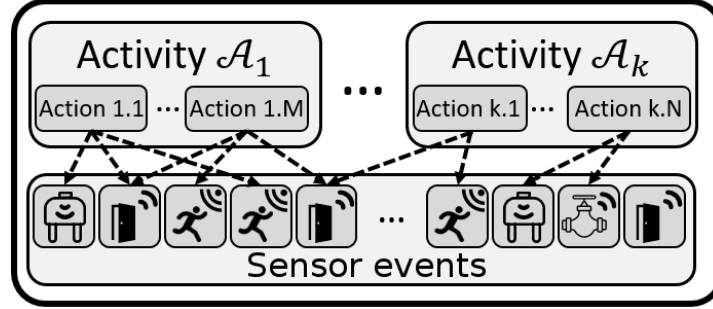


Figure 3.1.1: Graphical representation of the hierarchical decomposition of activities into actions linked to sensor events given by an expert.

In order to simplify the readability of equations and algorithms, the following notations are used:

- $\Sigma_{Actionk.h} = Link^{kAction/Event}(Actionk.h)$  is the set of events linked with the action  $Actionk.h$ ;
- $\Sigma_{\mathcal{A}_k}$  is the set of events linked with the activity  $\mathcal{A}_k$ . This set can be defined as 
$$\Sigma_{\mathcal{A}_k} = \bigcup_{action \in Link^{kActivity/Action}(\mathcal{A}_k)} Link^{kAction/Event}(action);$$
- $Le_{\mathcal{A}_k} = card(\Sigma_{\mathcal{A}_k})$  is the number of events hierarchically linked with the activity  $\mathcal{A}_k$ .

We are now able to give the definition of the used model type: the PFA. Probabilistic finite state automata is the name of a syntactic object (i.e. it contains symbols, lists, and constant values that essentially correspond to the quoted form of the expression) which attempts to model and generate probabilistic distributions over a set of possible infinite cardinality of strings or sequences. This object is defined as written in [Vidal et al. \(2005a\)](#) and in the following.

**Definition 3.4.** : A PFA  $\mathcal{A}_k$  is a tuple  $\mathcal{A}_k = \langle Q_{\mathcal{A}_k}, \Sigma_{\mathcal{A}_k}, \delta_{\mathcal{A}_k}, I_{\mathcal{A}_k}, F_{\mathcal{A}_k}, P_{\mathcal{A}_k} \rangle$ , where:

- $Q_{\mathcal{A}_k}$  is a finite non-empty set of states  $q$ ;
- $\Sigma_{\mathcal{A}_k}$  is a non-empty alphabet of events  $e$ ;
- $\delta_{\mathcal{A}_k} \subseteq Q_{\mathcal{A}_k} \times \Sigma_{\mathcal{A}_k} \times Q_{\mathcal{A}_k}$  is a set of transitions;
- $I_{\mathcal{A}_k} : Q_{\mathcal{A}_k} \rightarrow [0, 1]$  are the initial-state probabilities;
- $P_{\mathcal{A}_k} : \delta_{\mathcal{A}_k} \rightarrow [0, 1]$  are the transition probabilities;
- $F_{\mathcal{A}_k} : Q_{\mathcal{A}_k} \rightarrow [0, 1]$  are the final-state probabilities;



where  $I_{\mathcal{A}_k}$ ,  $P_{\mathcal{A}_k}$  and  $F_{\mathcal{A}_k}$  are functions such that:

$$\sum_{q \in Q_{\mathcal{A}_k}} I_{\mathcal{A}_k}(q) = 1, \quad (3.1)$$

and

$$\forall q \in Q_{\mathcal{A}_k}, F_{\mathcal{A}_k}(q) + \sum_{e \in \Sigma_{\mathcal{A}_k}, q' \in Q_{\mathcal{A}_k}} P_{\mathcal{A}_k}(q, e, q') = 1. \quad (3.2)$$

PFA are represented as directed labelled graphs. Figure 3.1.2 shows an example of a PFA with four states:  $Q_{\mathcal{A}_k} = \{q_0, q_1, q_2, q_3\}$ , with  $I_{\mathcal{A}_k}(q_0) = 1$  and  $I_{\mathcal{A}_k}(q_i) = 0$  for  $i \in \{1, 2, 3\}$ , and a four-symbol alphabet:  $\Sigma_{\mathcal{A}_k} = \{a, b, c, d\}$ . Each transition (resp. state) is labelled by the linked event (resp. the state name) and its possibility to occur (resp. its final probability).

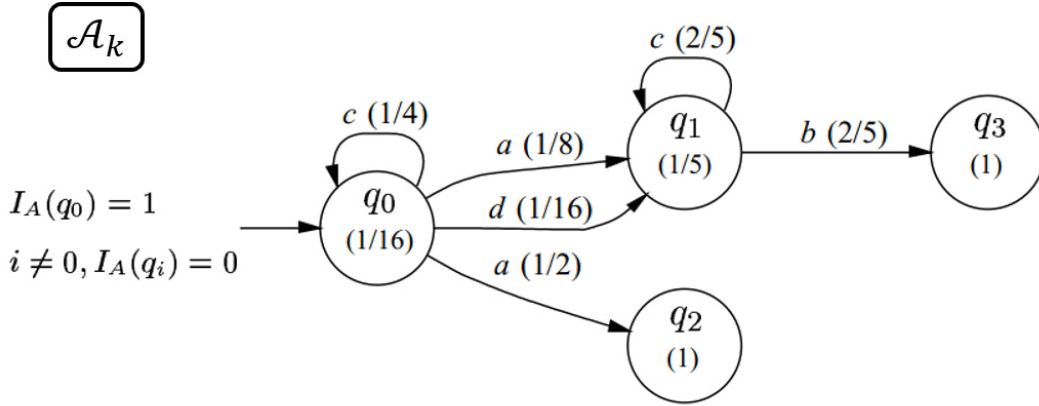


Figure 3.1.2: Graphical representation of a PFA  $\mathcal{A}_k$  (Vidal et al., 2005b).

In the following study, the term  $\mathcal{A}_k$  is used to represent both activity  $k$  and the PFA modelling this activity. When used in subscript of a symbol,  $\mathcal{A}_k$  represents the PFA to which the symbol is linked. Moreover, we call *structure of a PFA  $\mathcal{A}_k$*  the triplet  $structure(\mathcal{A}_k) = \langle Q_{\mathcal{A}_k}, \Sigma_{\mathcal{A}_k}, \delta_{\mathcal{A}_k} \rangle$  where all is known except probabilities.

In the given example, the transition  $(q_0, a, q_1)$  going from  $q_0$  to  $q_1$  and having a probability  $P_{\mathcal{A}_k}(q_0, a, q_1) = 1/8$  can be read as the probability, for the event generator modelled by the PFA  $\mathcal{A}_k$  (shown figure 3.1.2) to reach state  $q_1$  from state  $q_0$  by generating the event  $a$ .

According to [Cassandras and Lafortune \(2009\)](#), the key operation involved in building strings (or sequences), from a set of events  $\Sigma$  is *concatenation*. The string  $abb$  is the concatenation of the string  $ab$  with the event (or string of length one)  $b$ ;  $ab$  is itself the concatenation of  $a$  and  $b$ . The concatenation  $uv$  of two strings  $u$  and  $v$  is the new string consisting of the events in  $u$  immediately followed by the events in  $v$ . The empty string  $\epsilon$  is the identity element of concatenation:  $u\epsilon = \epsilon u = u$  for any string  $u$ .

Furthermore, by defining  $\Sigma^*$  as the Kleene-closure of the set of events  $\Sigma$ , i.e. the infinite set of all possible sequences of elements of  $\Sigma$ , we can define the projection function. This function is needed during our AD procedure in order to avoid noise influence and its definition and is the following:

**Definition 3.5** (Projection function). *The projection of  $w \in \Sigma^*$  on alphabet  $\Sigma_{\mathcal{A}_k}$  is noted  $Proj(w, \Sigma_{\mathcal{A}_k})$  and defined as defined in (Cassandras and Lafortune, 2009):*

$$Proj : \Sigma^* \rightarrow \Sigma_{\mathcal{A}_k}^* ;$$

with:

$$Proj(\epsilon, \Sigma_{\mathcal{A}_k}) = \epsilon ;$$

$$Proj(v, \Sigma_{\mathcal{A}_k}) = \begin{cases} v & \text{if } v \in \Sigma_{\mathcal{A}_k} \\ \epsilon & \text{else} \end{cases} ;$$

$$Proj(\sigma v, \Sigma_{\mathcal{A}_k}) = Proj(\sigma, \Sigma_{\mathcal{A}_k}) Proj(v, \Sigma_{\mathcal{A}_k}) \text{ for } \sigma \in \Sigma^*, v \in \Sigma ;$$

As can be seen from the definition, the projection operation takes a string formed from the larger event set ( $\Sigma$ ) and *erases* events in it that do not belong to the smaller event set ( $\Sigma_{\mathcal{A}_k}$ ).

Finally, the following notations are also chosen to simplify the approach formulation:

- $\Sigma_{q_l}$  the alphabet of events associated with the transitions ongoing into the state  $q_l$ ;
- $w^{Obs} = e_i e_j \dots e_n$  is a sequence of the observed events in the smart home. The length of  $w^{Obs}$  (denoted  $|w^{Obs}|$ ) corresponds to the number of events composing the sequence;
- $w_{p_k}^{Obs} = Proj(w^{Obs}, \Sigma_{\mathcal{A}_k})$  is the projection of the sequence  $w^{Obs}$  on the alphabet  $\Sigma_{\mathcal{A}_k}$ ;

## 3.2 A systematic procedure for models generation

In this section, the objective is to model each activity  $\mathcal{A}_k$  we want to monitor by a PFA. To this aim, the structure of the PFA is first generated in 3.2.1. Then probabilities are learned 3.2.3 thanks to a database splitting chosen in 3.2.2.

### 3.2.1 Generation of PFA structure

We know that for some methods (as in Oliver et al. (2002)) based on the Baum-Welch algorithm (Baum, 1972), the structure is not very important. Indeed, probabilities are computed iteratively without considering any semantic signification linked with the state name. However, those methods are efficient only if the sequences given to each iterative step are representative of the modelled activity.

Unfortunately, let's recall that according to Assumption 4, in our proposed method the knowledge of the performed activity is rejected. Thus, no representative sequences can easily be extracted from the learning database. Therefore, in order to deal with this information reject, an expert decomposition of activities into a hierarchy of actions and events is used to generate activity PFA structures, which consequently matters to us.

First, for each action  $Action_k.h \in Link_{Activity/Action}(\mathcal{A}_k)$  linked to the activity  $\mathcal{A}_k$ , we associate a state  $q_h$ . Then, a unique initial state  $q_0$  not linked to any action is added (as shown in figure 3.2.1). At this point,  $Q_{\mathcal{A}_k} = \{Link_{Activity/Action}(\mathcal{A}_k) + q_0\}$ .

The objective is now to illustrate the links between the  $Le_{\mathcal{A}_k}$  events and the different actions in the expert decomposition. We recall that an event of the smart home can be

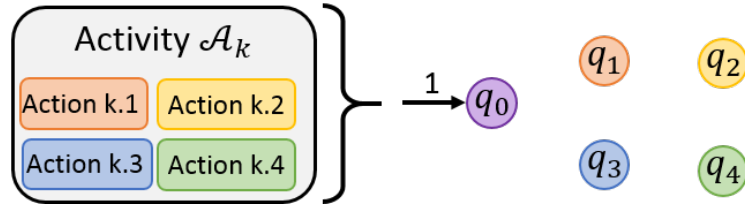


Figure 3.2.1: Example of the creation of a set of five states  $Q_{\mathcal{A}_k} = \{q_0, q_1, q_2, q_3, q_4\}$  from an activity  $\mathcal{A}_k$  decomposed into four actions (Action  $k.1$  to  $k.4$ ).

linked to zero, one or several actions of the same activity  $\mathcal{A}_k$ . Only events linked with at least one action of  $\mathcal{A}_k$  are counted in  $Le_{\mathcal{A}_k}$ .

Thus, if we want to illustrate the capacity of an action  $Actionk.h$  to generate an event  $e_i$  during its performance (e.g. the ability to generate an event "Open the frigde" when a ready-cook dish is prepared), a transition  $(q_g, e_i, q_h)$  is added to the structure for each couple of actions  $(q_g, q_h)$  and the event  $e_i$  is added to the alphabet  $\Sigma_{\mathcal{A}_k}$  (see figure 3.2.2). Here, the state  $q_h$  is the state representing the action  $Actionk.h$  and  $q_g$  any state of the model.

At this step, with  $card(Q_{\mathcal{A}_k})$  the number of states previously created, at most  $card(Q_{\mathcal{A}_k}) \times Le_{\mathcal{A}_k}$  transitions are generated.

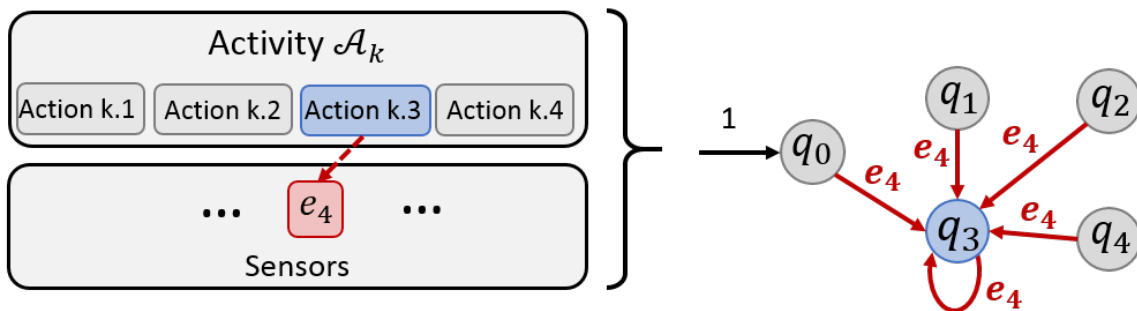


Figure 3.2.2: Example of the creation of five transitions  $\{(q_0, e_4, q_3), (q_1, e_4, q_3), (q_2, e_4, q_3), (q_3, e_4, q_3), (q_4, e_4, q_3)\}$  reflecting the link between the action  $N.3$  and the event  $e_4$

The structure, and thus the model, generated there has to be seen as an interpreter. If we are incoming into a state  $q_g$  linked with a given action, it signifies that we are starting to perform this action. If we are staying in this state, by passing through self-loops, we are carrying on the action.

At the end of this step, the structure of each PFA associated with an activity  $\mathcal{A}_k$  is fully known. In order to improve the figure readability, states are labelled by the corresponding action name and  $q_0$  is labelled by "Initial". Fig. 3.2.3 shows a generic overview of this structure generation starting from the expert decomposition.

The algorithm 1 gives a pseudo-code representation of this step. By analysing it, we can observe that the complexity  $C_{AD_1}$  of this AD step 1 is polynomial and can be

expressed as follows:

$$C_{AD_1} = O \left( \text{card}(lActivities) \times \left( \max_{\mathcal{A}_k} \left( \text{card}(Link_{Activity/Action}(\mathcal{A}_k)) \right) \right)^2 \times \max_{\mathcal{A}_k} (Le_{\mathcal{A}_k}) \right) \quad (3.3)$$

with:

- $\text{card}(lActivities)$  the number of activities to monitor;
- $\max_{\mathcal{A}_k} \left( \text{card}(Link_{Activity/Action}(\mathcal{A}_k)) \right)$  the maximum number of actions links to the same activity in the expert decomposition;
- $\max_{\mathcal{A}_k} (Le_{\mathcal{A}_k})$  the maximum number of events; hierarchically linked to the same activity in the expert decomposition. It maximises the number of events hierarchically linked to the same action.

---

**Algorithm 1** AD step 1: Generation of a PFA structure

---

**Require:**

- A list of activities  $lActivities : \{\mathcal{A}_k\}$  to monitor;
- A hierarchical decomposition activities/actions:  
 $Link_{Activity/Action}(\mathcal{A}_k) : \{action\ k.1, action\ k.2, \dots\}$ ;
- A hierarchical decomposition actions/events:  
 $\Sigma_{action\ k.h} : \{e_i, \dots, e_j\}$

**Ensure:** A set of PFA structures  $Structures$ , one for each activity  $\mathcal{A}_k$ .

```

1: // Global Initialisation:
2: Structures := ∅
3: // For loop over all activities:
4: for all  $\mathcal{A}_k \in lActivities$  do
5:    $Q_{\mathcal{A}_k} := \{Initial, \}$ 
6:    $\Sigma_{\mathcal{A}_k} := \emptyset$ 
7:    $\delta_{\mathcal{A}_k} := \emptyset$ 
8:   // States creation:
9:   for all  $action \in Link_{Activity/Action}(\mathcal{A}_k)$  do
10:    append  $action$  to  $Q_{\mathcal{A}_k}$ 
11:   end for
12:   // Transitions and alphabet creation:
13:   for all  $end\_action \in Q_{\mathcal{A}_k} - Initial$  do
14:     for all  $begin\_action \in Q_{\mathcal{A}_k}$  do
15:       for all  $event \in \Sigma_{end\_action}$  do
16:         append  $(begin\_action, event, end\_action)$  to  $\delta_{\mathcal{A}_k}$ 
17:         if  $event \notin \Sigma_{\mathcal{A}_k}$  then
18:           append  $event$  to  $\Sigma_{\mathcal{A}_k}$ 
19:         end if
20:       end for
21:     end for
22:   end for
23:   append  $\langle Q_{\mathcal{A}_k}, \Sigma_{\mathcal{A}_k}, \delta_{\mathcal{A}_k} \rangle$  to  $Structures$ 
24: end for

```

---

At this point, all sequences  $w \in \Sigma_{\mathcal{A}_k}^*$  composed with events hierarchically linked with the activity  $\mathcal{A}_k$  are generated by the PFA structure. The constructed structure is non-

deterministic since, starting from a state  $q_g$ , several transition generating the same event  $e_i$  and outgoing to different states are existing. This indeterminism expresses the choice to not promote any hypothesis on the compartment at this step of the discovery. This kind of choice is thus reported to the probability computation and will be commented at this step.

Once the PFA structures generated, the probabilities linked to each transition have to be computed. Steps 3.2.2 and 3.2.3 explain the proposed solution.

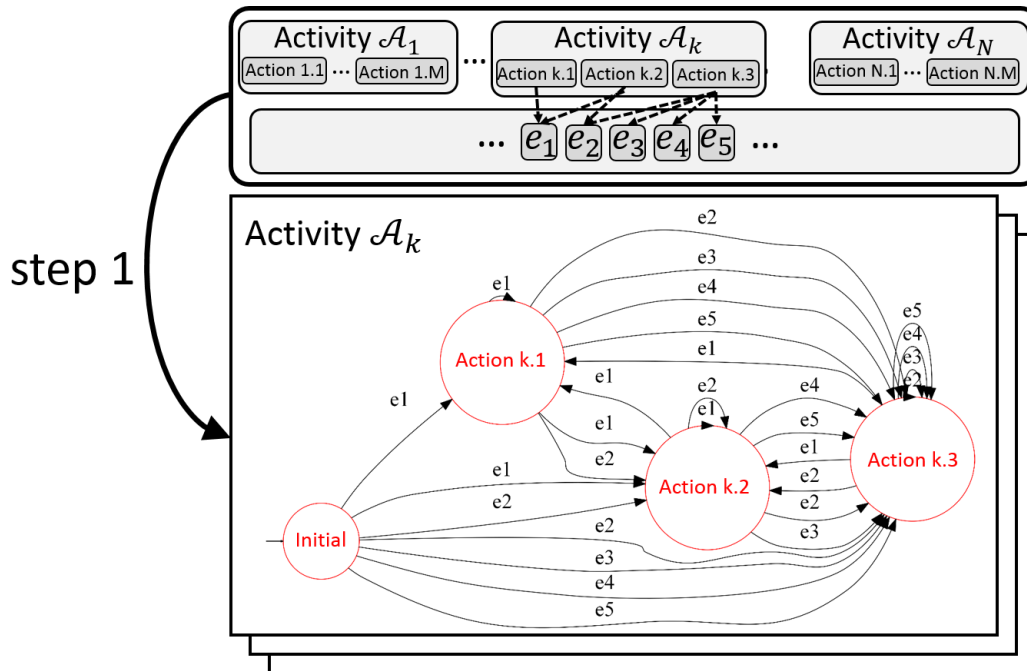


Figure 3.2.3: From expert decomposition to PFA structure.

### 3.2.2 Database of event logs exploration

As the PFA structure is now known, we need to get the probabilities of each transition  $(q_g, e_i, q_h)$ . To compute these values, we use a learning approach by observing events during a learning period. Specific key indicators that can be extracted from this observation help us to compute probabilities. Those indicators, described in the next subsections, correspond to the count of some situation occurrences in the recorded event database. The database consists of one or several long sequences of recorded events in which we have to look for local succession of actions. These long sequences of events obtained thanks to observation is usually divided into smaller segments in order to reduce the needed computational resources. In the existing literature (Cook and Krishnan, 2015; Oliver et al., 2002), indicators values are obtained from several well-chosen sequences extracted from the learning database using a log of performed activities. Once again in this work, the Assumption 4 excludes the use of this kind of log considered as a non-suitable input of the problem and another way to divide the learning database has to be used.

It exists different ways to divide such a huge sequence in several ones. The most common approach is to split the sequence by considering a sliding subsequence of events. In Krishnan and Cook (2014), authors present three common approaches for processing such subsequence of events (see Figure 3.2.4):

- (i) explicit segmentation;
- (ii) time based windowing;
- (iii) sensor event based windowing.

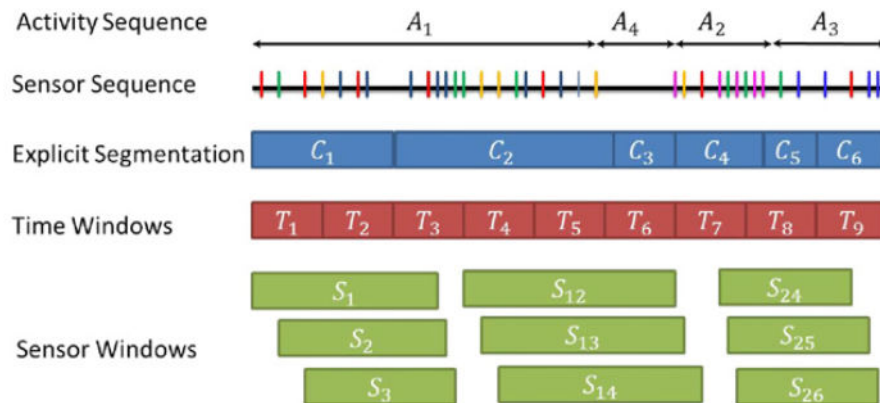


Figure 3.2.4: Common approaches for processing streaming data (from (Krishnan and Cook, 2014))

#### Explicit Segmentation

In Junker et al. (2008), the streaming sensor events are segmented into chunks, each chunk possibly corresponding to an activity. An example of this process is illustrated in Figure 3.2.4. The segmentation process results in chunks  $C_1, C_2, \dots, C_6$  obtained by processing a naive activity recognition process. We can see that the process does not result in exact activity boundaries (activity  $A_1$  is broken down to chunks  $C_1$  and  $C_2$ ). This fact is common with all other segmentation algorithms. While the segmentation

process could lead to chunks representing specific activities, it has some drawbacks. The foremost one is that the approach has to wait for future data to make a decision on past data rendering it a somewhat non-streaming approach. Furthermore, because of its dependency on the future data, large temporal gaps in successive sensor events, which are realistic in everyday routines, will result in the approach waiting for a long time to make a decision on the past events. Finally, this approach also leads to additional complexity issues of the segmentation process to be dealt with such as splitting and merging chunks. This segmentation approach is particularly adapted with activities having very distinct boundaries. Unfortunately this is not the usual scenario.

### Time windows

The second approach to handle streaming data is to divide the entire sequence of sensor events into equal size time intervals as illustrated in Figure 3.2.4 by the chunks denoted by  $T_1, T_2, \dots, T_9$ . This approach has been adopted by many researchers (Krishnan and Panchanathan, 2008; Van Kasteren et al., 2010). This technique offers a simple approach to learn the activity models during the training phase over the explicit segmentation approach. It further reduces the computational complexity of the explicit segmentation process. This is a good approach when dealing with data obtained from periodic generation of events. Data for every time interval is always guaranteed in such a scenario. This is a common approach with accelerometers and gyroscopes, where data is sampled at a constant rate from the sensors. However, one has to deal with the problem of selecting the optimal length of the time interval. If a very small interval is chosen, there is a possibility that it will not contain any relevant activity information. If the time interval is too large, information belonging to multiple activities can be embedded into it. Hence, the activity that dominates the time interval will have a greater influence in the classification decision. This problem manifests itself when dealing with sensors that do not have a constant sampling rate. In the current context of motion and door sensor events, it is very likely that some time intervals do not have any sensor events in them (e.g.,  $T_6$  in Figure 3.2.4). Then heuristics have to be developed to extend the activity occurring in the previous time intervals to the current time interval.

### Sensor Windows

The third approach for sensor stream processing is to divide the sequence into windows containing an equal number of sensor events. This is illustrated in figure 3.2.4 by the chunks  $S_1, S_2, \dots, S_{26}$ . It is evident that the windows appear to vary in their duration. This is fine considering that during the performance of activities, multiple sensors could be triggered, while during silent periods, there will not be many sensor firings. The sensor events preceding the last event in a window define the context for the last event. This method also has some inherent drawbacks. For example, consider the chunk  $S_{13}$ , in Figure 3.2.4. The last sensor event of this chunk corresponds to the beginning sensor event of activity  $A_2$ . There is a significant time lag between this event and its preceding sensor event. The relevance of all the sensor events in this chunk on the last event might be small if the time lag is large. While by itself this approach may not be alluring, modifying it to account for the relationship between the sensor events is a good method to process the stream of sensor events. This approach offers computational advantages over the explicit segmentation process and does not require future sensor events for classifying past sensor events.

Therefore, sensor windowing is preferred in this work since we have activities with a variable duration and separated with a variable delay. Furthermore, let's recall that the idea behind the dataset splitting is to compute some indicators traducing the event

sensors relationship. Those indicators can thus be used to compute the model probabilities by learning as presented in the next subsection. However the algorithm 2 giving a pseudo-code representation of the complete step 2 is presented here. By analysing it, we can observe that the time complexity  $C_{AD_2}$  of this AD step 2 is linear and can be expressed as follows:

$$C_{AD_2} = O\left(\left(|LearningDB| - |w^{Obs}|\right) \times card(Structures) \times |w^{Obs}|\right) \quad (3.4)$$

with:

- $\left(|LearningDB| - |w^{Obs}|\right)$  the number of windows obtained by sliding;
- $card(Structures) = card(lActivities)$  the number of activities to monitor;
- $|w^{Obs}|$  the length of the learning windows.

### 3.2.3 Probabilities computation

As precise in the PFA definition in subsection 3.1, each transition  $(q_g, e_i, q_h)$  represents the fact that, during the activity performance, the inhabitant performs successively the action  $q_g$  then  $q_h$  and that the first event of the action  $q_h$  generated by this transition is the event  $e_i$ . Therefore, probabilities  $P_{\mathcal{A}_k}(q_g, e_i, q_h)$  linked to transition  $(q_g, e_i, q_h)$  represents the probability, for an inhabitant performing the activity  $\mathcal{A}_k$  to generate the event  $e_i$  by changing from action  $q_g$  to action  $q_h$ .

In order to compute those probabilities, it is first necessary to avoid the influence of noise on the computation precision. To this aim, for each activity, a projection of the sliding window  $w^{Obs}$  presented above (subsection 3.2.2 and in figure 3.2.4) on the activity alphabet  $\Sigma_{\mathcal{A}_k}$  is performed. The obtained sequence  $w_{p_k}^{Obs}$  is, as defined in subsection 3.1, generated by removing the events  $e_i \notin \Sigma_{\mathcal{A}_k}$  from  $w^{Obs}$ .

The transition  $(q_g, e_i, q_h)$  can be seen as a consequence of two different and independent human decisions:

- the choice to perform the action  $q_h$  after the action  $q_g$ ;
- the choice to perform a move generating event  $e_i$  as first observable move following the action change decision.

Probabilistically thinking, this point of view is equivalent to decompose the probability  $P(q_g, e_i, q_h)$  as the product of two probabilities: the probability to move from action  $q_g$  to action  $q_h$  and the probability to generate  $e_i$  during this change. Thus, we have:

$$P(q_g, e_i, q_h) = P(q_g \rightarrow q_h | q_g) \times P(e_i | q_g \rightarrow q_h) \quad (3.5)$$

By supposing than we have enough knowledge to directly count changes linked with each probability and that  $N(q_g \rightarrow q_h | q_g)$  and  $N(e_i | q_g \rightarrow q_h)$  denote the number of occurrences of transitions from  $q_g$  to  $q_h$  and the occurrences of event  $e_i$  conditioned to the transition  $q_l \rightarrow q_m$  during the run period, respectively, we know that:

$$P(q_g \rightarrow q_h | q_g) = \frac{N(q_g \rightarrow q_h | q_g)}{\sum_{q_m \in Q_{\mathcal{A}_k}} N(q_g \rightarrow q_m | q_g)} \quad (3.6)$$

$$P(e_i | q_g \rightarrow q_h) = \frac{N(e_i | q_g \rightarrow q_h)}{\sum_{e_j \in \Sigma_{\mathcal{A}_k}} N(e_j | q_g \rightarrow q_h)} \quad (3.7)$$



**Algorithm 2** AD step 2: Database of event logs exploration**Require:**

- A set of PFA Structures  
 $Structures : \{ \langle Q_{\mathcal{A}_1}, \Sigma_{\mathcal{A}_1}, \delta_{\mathcal{A}_1} \rangle \dots \langle Q_{\mathcal{A}_k}, \Sigma_{\mathcal{A}_k}, \delta_{\mathcal{A}_k} \rangle \};$
- A learning database of events  $LearningDB = [e_1, e_5, \dots, e_i, \dots, e_j];$
- A sliding window length  $|w^{Obs}|.$

**Ensure:**

- The dictionary  $dictOccurs$  with keys  $(\mathcal{A}_k, e_i)$  and values representing  $\{N_{e_i}^k\};$
  - The dictionary  $dictInits$  with keys  $(\mathcal{A}_k, e_i)$  and values representing  $\{N_{init\ e_i}^k\};$
  - The dictionary  $dictFollows$  with keys  $(\mathcal{A}_k, e_i, e_j)$  and values representing  $\{N_{e_i \rightarrow e_j}^k\}.$
- 1: // Global Initialisation:
  - 2:  $dictOccurs := \emptyset$
  - 3:  $dictInits := \emptyset$
  - 4:  $dictFollows := \emptyset$
  - 5: // For all windows of length  $|w^{Obs}|$  in the learning database  $|LearningDB|:$
  - 6: **for**  $index := 1$  **to**  $|LearningDB| - |w^{Obs}| + 1$  **do**
  - 7:    $w^{Obs} := LearningDB[index\ to\ index + |w^{Obs}| - 1]$
  - 8:   // For all activity structure (and by extension, for all activities):
  - 9:   **for all**  $\langle Q_{\mathcal{A}_k}, \Sigma_{\mathcal{A}_k}, \delta_{\mathcal{A}_k} \rangle \in Structures$  **do**
  - 10:      $w_{p_k}^{Obs} = Proj(w^{Obs}, \Sigma_{\mathcal{A}_k})$
  - 11:     //  $N_{init\ e_i}^k$  incrementation:
  - 12:     **if**  $key(\mathcal{A}_k, w_{p_k}^{Obs}[1]) \in dictInits$  **then**
  - 13:        $dictInits[(\mathcal{A}_k, w_{p_k}^{Obs}[1])] := dictInits[(\mathcal{A}_k, w_{p_k}^{Obs}[1])] + 1$
  - 14:     **else**
  - 15:        $dictInits[(\mathcal{A}_k, w_{p_k}^{Obs}[1])] := 1$
  - 16:     **end if**
  - 17:     //  $N_{e_i}^k$  incrementation:
  - 18:     **for all** event  $e_i \in w_{p_k}^{Obs}$  **do**
  - 19:       **if**  $key(\mathcal{A}_k, e_i) \in dictOccurs$  **then**
  - 20:           $dictOccurs[(\mathcal{A}_k, e_i)] := dictInits[(\mathcal{A}_k, e_i)] + 1$
  - 21:       **else**
  - 22:           $dictOccurs[(\mathcal{A}_k, e_i)] := 1$
  - 23:       **end if**
  - 24:     **end for**
  - 25:     //  $N_{e_i \rightarrow e_j}^k$  incrementation:
  - 26:     **for**  $index\_couples := 1$  **to**  $|w_{p_k}^{Obs}| - 2 + 1$  **do**
  - 27:        $e_i := w_{p_k}^{Obs}[index\_couples]$
  - 28:        $e_j := w_{p_k}^{Obs}[index\_couples + 1]$
  - 29:       **if**  $key(\mathcal{A}_k, e_i, e_j) \in dictFollows$  **then**
  - 30:           $dictFollows[(\mathcal{A}_k, e_i, e_j)] := dictFollows[(\mathcal{A}_k, e_i, e_j)] + 1$
  - 31:       **else**
  - 32:           $dictFollows[(\mathcal{A}_k, e_i, e_j)] := 1$
  - 33:       **end if**
  - 34:     **end for**
  - 35:   **end for**
  - 36: **end for**

Unfortunately, due to the same reasons as those explained in chapter 1 leading us to reject the performed activities knowledge as a reliable input, we cannot consider the more detailed performed action knowledge as an input. Thus, the values of  $N(q_g \rightarrow q_h|q_g)$  and  $N(e_i|q_g \rightarrow q_h)$  have to be estimated using observable occurrences or successions of events. The retained countable indicators are the following and figure 3.2.5 represents the complete procedure from the windowing to the indicators count.

- $N_{e_i}^k$ : number of times where event  $e_i$  is observed in the sequence  $w_{p_k}^{Obs}$ ;
- $N_{init\ e_i}^k$ : number of times where event  $e_i$  is the first event of the projected sequence  $w_{p_k}^{Obs}$ ;
- $N_{e_i \rightarrow e_j}^k$ : number of times where event  $e_j$  follows event  $e_i$  in the projected sequence  $w_{p_k}^{Obs}$ .

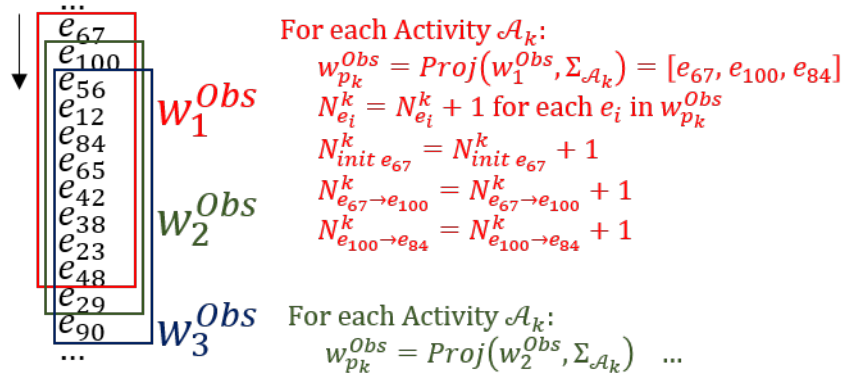


Figure 3.2.5: Sliding sensor window and indicator computation

Using these counters, the values of  $N(q_g \rightarrow q_h|q_g)$  and  $N(e_i|q_g \rightarrow q_h)$  are estimated as follows:

$$\tilde{N}(q_g \rightarrow q_h|q_g) = \begin{cases} \sum_{e_i \in \Sigma_{q_h}} \frac{1}{C_{e_i}} N_{init\ e_i}^k & \text{if } q_g = q_0 \\ \sum_{e_i, e_j \in \Sigma_{q_h}} N_{e_i \rightarrow e_j}^k & \text{if } q_g = q_h \neq q_0 \\ \sum_{e_i \in \Sigma_{q_g}; e_j \in \Sigma_{q_h}; e_j \notin \Sigma_{q_g}} \frac{1}{C_{e_j}} N_{e_i \rightarrow e_j}^k & \text{if } q_g \neq q_h \neq q_0 \end{cases} \quad (3.8)$$

and

$$\tilde{N}(e_i|q_g \rightarrow q_h) = N_{e_i}^k \text{ if } e_i \in \Sigma_{q_h}; 0 \text{ otherwise,} \quad (3.9)$$

where  $C_{e_j}$  is the number of states hierarchically linked in input with  $e_j$ .

When a human starts a new action, the performed move has no reason to be dependant with the past action. Therefore, in equation (3.9), the occurrence of  $e_i$  is independent from the starting action  $q_g$  but depends only on the reached action  $q_h$ .

In addition, when an event linked to two actions are observed, two cases can be considered. First, the human is changing action. Secondly, the human continue the current performed action. But, since the first observable move can be linked to the

previous action as well as the possible new one, a choice has to be done. In equation (3.8), the choice to count the event occurrence as linked to the previous action is done.

In equation (3.8), three cases are distinguished:

1. if  $q_g = q_0$ , initial indicators  $N_{init\ e_i}^k$  are considered,
2. if  $q_g = q_h$  then only succession of two events linked to  $q_g$  are considered as precise before,
3. if  $q_g \neq q_h$  then the succession of two events both linked to  $q_g$  and  $q_h$  is not considered.

The factor  $\frac{1}{C_{e_j}}$  shares each occurrence of event  $e_j$  between all states possibly concerned.

Finally, using the automatically computed frequency indicators  $N_{e_i}^k$ ,  $N_{init\ e_i}^k$  and  $N_{e_i \rightarrow e_j}^k$  the probabilities  $P(q_g, e_i, q_h)$  are defined as following:

$$P(q_g, e_i, q_h) = P(q_g \rightarrow q_h | q_g) \times P(e_i | q_g \rightarrow q_h)$$

with:

$$P(q_g \rightarrow q_h | q_g) = \frac{\tilde{N}(q_g \rightarrow q_h | q_g)}{\sum_{q_n \in Q_{\mathcal{A}_k}} \tilde{N}(q_g \rightarrow q_n | q_g)} \quad (3.10)$$

$$P(e_i | q_g \rightarrow q_h) = \frac{\tilde{N}(e_i | q_g \rightarrow q_h)}{\sum_{e_j \in \Sigma_{\mathcal{A}_k}} \tilde{N}(e_j | q_g \rightarrow q_h)}$$

and  $N(q_g \rightarrow q_h | q_g)$  (resp.  $N(e_i | q_g \rightarrow q_h)$ ) as defined in (3.8) (resp. (3.9))

At this point, for each activity  $\mathcal{A}_k$ , we define:

$$I_{\mathcal{A}_k}(q_h) = 1 \text{ if } q_h = q_0; 0 \text{ otherwise;} \quad (3.11)$$

and  $P_{\mathcal{A}_k} = \{P(q_g, e_i, q_h) \forall (q_g, e_i, q_h) \in \delta_{\mathcal{A}_k}\}$ .

As the inhabitant of a smart home can decide to stop an activity at any time and anywhere, define a final probability is nonsense. Therefore, in our models,  $F_{\mathcal{A}_k} = \emptyset$ .

The algorithm 3 gives a pseudo-code representation of this step. By analysing it, we can observe that the time complexity  $C_{AD_3}$  of this AD step 3 is polynomials considering the number of actions linked to activities, linear considering the rest and can be expressed as follows:

$$C_{AD_3} = O \left( \text{card}(\text{Structures}) \times \max_{\mathcal{A}_k} (\text{card}(Q_{\mathcal{A}_k})) \times \left( \max_{\mathcal{A}_k} (\text{card}(Q_{\mathcal{A}_k})) + \max_{\mathcal{A}_k} (Le_{\mathcal{A}_k}) \right) \right) \quad (3.12)$$

with:

- $\text{card}(\text{Structures}) = \text{card}(l\text{Activities})$  the number of activities to monitor;
- $\max_{\mathcal{A}_k} (\text{card}(Q_{\mathcal{A}_k}))$  the maximum number of actions links to the same activity;
- $\max_{\mathcal{A}_k} (Le_{\mathcal{A}_k})$  the maximum number of events linked to the same activity.

---

**Algorithm 3** AD step 3: Probabilities computation
 

---

**Require:**

- A set of PFA Structures  
 $Structures : \{ \langle Q_{\mathcal{A}_1}, \Sigma_{\mathcal{A}_1}, \delta_{\mathcal{A}_1} \rangle \dots \langle Q_{\mathcal{A}_k}, \Sigma_{\mathcal{A}_k}, \delta_{\mathcal{A}_k} \rangle \};$
- Three dictionaries of counters:
  - The dictionary  $dictOccurs$  with keys  $(\mathcal{A}_k, e_i)$  and values  $\{N_{e_i}^k\}$ ;
  - The dictionary  $dictInits$  with keys  $(\mathcal{A}_k, e_i)$  and values  $\{N_{init\ e_i}^k\}$ ;
  - The dictionary  $dictFollows$  with keys  $(\mathcal{A}_k, e_i, e_j)$  and values  $\{N_{e_i \rightarrow e_j}^k\}$ .

**Ensure:** A set of PFAs  $PFAs : \{\mathcal{A}_k\}$  modelling the activities to monitor

```

1: //Global Initialisation:
2:  $PFAs := \emptyset$ 
3: // For all activity structures (and by extension, for all activities):
4: for all  $\langle Q_{\mathcal{A}_k}, \Sigma_{\mathcal{A}_k}, \delta_{\mathcal{A}_k} \rangle \in Structures$  do
5:    $I_{\mathcal{A}_k}(Initial) := 1$ 
6:   for all  $state \in Q_{\mathcal{A}_k} - Initial$  do
7:      $I_{\mathcal{A}_k}(state) := 0$ 
8:   end for
9:    $F_{\mathcal{A}_k} := \emptyset$ 
10:   $P_{\mathcal{A}_k} := \emptyset$  (dictionary of probabilities)
11:  // Computation of the different  $\tilde{N}$  needed:
12:  for all  $end\_action \in Q_{\mathcal{A}_k} - Initial$  do
13:    for all  $begin\_action \in Q_{\mathcal{A}_k}$  do
14:      if  $begin\_action = Initial$  then
15:         $\tilde{N}(Initial \rightarrow end\_action) := \sum_{e_i \in \Sigma_{end\_action}} \frac{1}{C_{e_i}} N_{init\ e_i}^k$ 
16:      else if  $begin\_action = end\_action$  then
17:         $\tilde{N}(end\_action \rightarrow end\_action) := \sum_{e_i, e_j \in \Sigma_{end\_action}} N_{e_i \rightarrow e_j}^k$ 
18:      else

```

---

By applying this probability computation for each activity to monitor, the AD is fully performed. Indeed, at the end of the probability computation, we have a complete PFA  $\mathcal{A}_k = \langle Q_{\mathcal{A}_k}, \Sigma_{\mathcal{A}_k}, \delta_{\mathcal{A}_k}, I_{\mathcal{A}_k}, F_{\mathcal{A}_k}, P_{\mathcal{A}_k} \rangle$  modelling each activity with  $Q_{\mathcal{A}_k}, \Sigma_{\mathcal{A}_k}, \delta_{\mathcal{A}_k}$  defined in subsection 3.2.1 and  $I_{\mathcal{A}_k}, F_{\mathcal{A}_k}, P_{\mathcal{A}_k}$  defined in this subsection 3.2.3.

Finally, by applying the three steps of AD models building, the complexity  $C_{AD}$  of the proposed method is:

$$C_{AD} = C_{AD_1} + C_{AD_2} + C_{AD_3}; \quad (3.13)$$

Thus we have:

$$\begin{aligned}
 C_{AD} &= O \left( card(lActivities) \times \left( \max_{\mathcal{A}_k} (card(Link_{Activity/Action}(\mathcal{A}_k))) \right)^2 \times \max_{\mathcal{A}_k} (Le_{\mathcal{A}_k}) \right) \\
 &+ O \left( (|LearningDB| - |w^{Obs}|) \times card(Structures) \times |w^{Obs}| \right) \\
 &+ O \left( card(Structures) \times \max_{\mathcal{A}_k} (card(Q_{\mathcal{A}_k})) \times \left( \max_{\mathcal{A}_k} (card(Q_{\mathcal{A}_k})) + \max_{\mathcal{A}_k} (Le_{\mathcal{A}_k}) \right) \right)
 \end{aligned} \quad (3.14)$$

In practice, the number of actions linked to an activity ( $card(Q_{\mathcal{A}_k})$ ) can be considered as negligible compared to the number of events linked to the same activity ( $Le_{\mathcal{A}_k}$ ). Thus,

---

```

19:       $\tilde{N}(\text{begin\_action} \rightarrow \text{end\_action}) := \sum_{\substack{e_i \in \Sigma_{\text{begin\_action}} \\ e_i \notin \Sigma_{\text{end\_action}} \\ e_j \in \Sigma_{\text{end\_action}}}} \frac{1}{C_{e_j}} N_{e_i \rightarrow e_j}^k$ 
20:      end if
21:      end for
22:      for all  $\text{event} \in \Sigma_{\mathcal{A}_k}$  do
23:          if  $\text{event} \in \Sigma_{\text{end\_action}}$  then  $\tilde{N}(\text{event}|\text{end\_action}) := \text{dictOccurs}[(\mathcal{A}_k, \text{event})]$ 
24:          else  $\tilde{N}(\text{event}|\text{end\_action}) := 0$ 
25:          end if
26:      end for
27:  end for
28:  // Computation of the probabilities:
29:  for all  $\text{begin\_action} \in Q_{\mathcal{A}_k}$  do
30:       $\text{temporary\_sum\_1} := \sum_{q_n \in Q_{\mathcal{A}_k}} \tilde{N}(\text{begin\_action} \rightarrow q_n)$ 
31:      for all  $\text{end\_action} \in Q_{\mathcal{A}_k} - \text{Initial}$  do
32:           $P(\text{begin\_action} \rightarrow \text{end\_action}) := \frac{\tilde{N}(\text{begin\_action} \rightarrow \text{end\_action})}{\text{temporary\_sum\_1}}$ 
33:      end for
34:  end for
35:  for all  $\text{end\_action} \in Q_{\mathcal{A}_k} - \text{Initial}$  do
36:       $\text{temporary\_sum\_2} := \sum_{e_j \in \Sigma_{\mathcal{A}_k}} \tilde{N}(e_j|\text{end\_action})$ 
37:      for all  $\text{event} \in \Sigma_{\mathcal{A}_k}$  do
38:           $P(\text{event}|\text{end\_action}) := \frac{\tilde{N}(\text{event}|\text{end\_action})}{\text{temporary\_sum\_2}}$ 
39:      end for
40:  end for
41:  for all  $(\text{begin\_action}, \text{event}, \text{end\_action}) \in \delta_{\mathcal{A}_k}$  do
42:       $P_{\mathcal{A}_k}[(\text{begin\_action}, \text{event}, \text{end\_action})] := P(\text{begin\_action} \rightarrow \text{end\_action}) \times$   

 $P(\text{event}|\text{end\_action})$ 
43:  end for
44: end for

```

---

the complexity of the three AD is polynomial and can be approximated to:

$$\begin{aligned}
 C_{AD} &= O \left( \left[ (|\text{LearningDB}| - |w^{Obs}|) \times |w^{Obs}| + \left( \max_{\mathcal{A}_k} (\text{card}(Q_{\mathcal{A}_k})) \right)^2 + \max_{\mathcal{A}_k} (Le_{\mathcal{A}_k}) \right] \right. \\
 &\quad \left. \times \text{card}(l\text{Activities}) \right)
 \end{aligned} \tag{3.15}$$

In order to illustrate the presented AD method, an application is made in the next section.

### 3.3 Application to the Case Study

In this section, the presented AD method is applied to the case study presented in section 2.2 and the generation of the three activity models are presented and discussed.

#### 3.3.1 Generation of PFA structure

In order to generate PFAs structures, we start from the expert hierarchical decomposition of activities into actions then into events given in figure 2.2.10 and recall in figure 3.3.1. In the proposed method, the activities to be monitored are fixed by the medical staff and the expert instruments smart homes specifically to observe the activities. Additionally, the expert completes the medical knowledge by linking the actions composing the activities and the implemented sensors.

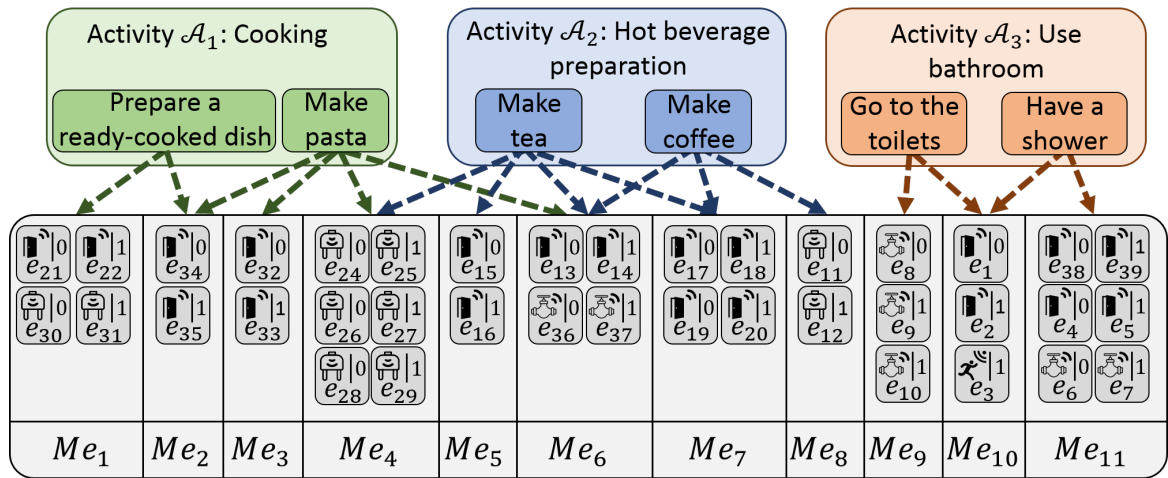


Figure 3.3.1: Expert hierarchical decomposition of activities to monitor and merge groups  $Me_i$  for representation

In this case study, we have to generate three PFAs  $\mathcal{A}_1, \mathcal{A}_2$  and  $\mathcal{A}_3$  modelling the three activities  $\mathcal{A}_1$ : cooking,  $\mathcal{A}_2$ : hot beverage preparation and  $\mathcal{A}_3$ : use bathroom. As recalled in the previous section, activities are decomposed in actions (two for each activity) themselves linked to events. Some events are shared between actions of the same activity as well as between activities. As our AD procedure treats independently each activity, the phenomenon of shared events between activities will not influence the aspect of our PFA structures. Thus, if the expert decides to modify its hierarchical decomposition for a unique activity (e.g. due to a new equipment implementation), the learning can be re-performed for this activity only without to change other models.

Figures 3.3.3 shows the structure generation of the activities  $\mathcal{A}_1$ . The structure generations of the activities  $\mathcal{A}_2$  and  $\mathcal{A}_3$  are reported in appendix B. For a readability issue of the intermediate models in the figures, events linked with the same actions are merged to label the same arrow. This merge is a graphical simplification only. At the end of this step, only the developed structured is kept. In the probability computation step, a different probability will be associated with each event and thus, this graphical merge will not be applicable any more. The made *merge group*  $Me_i$  are visible in figure 3.3.1 and the figure 3.3.2 shows how to read this graphical simplification on a simple example involving the merge group  $Me_1$ . The complete final structure is represented in the last cell of each explicative figures.

At this point, the structures of the three activities  $\mathcal{A}_1$ : *Cooking*,  $\mathcal{A}_2$ : *Hot beverage preparation* and  $\mathcal{A}_3$ : *Use bathroom* are fully known.

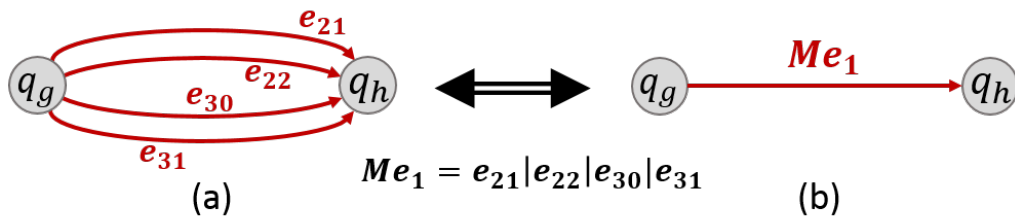
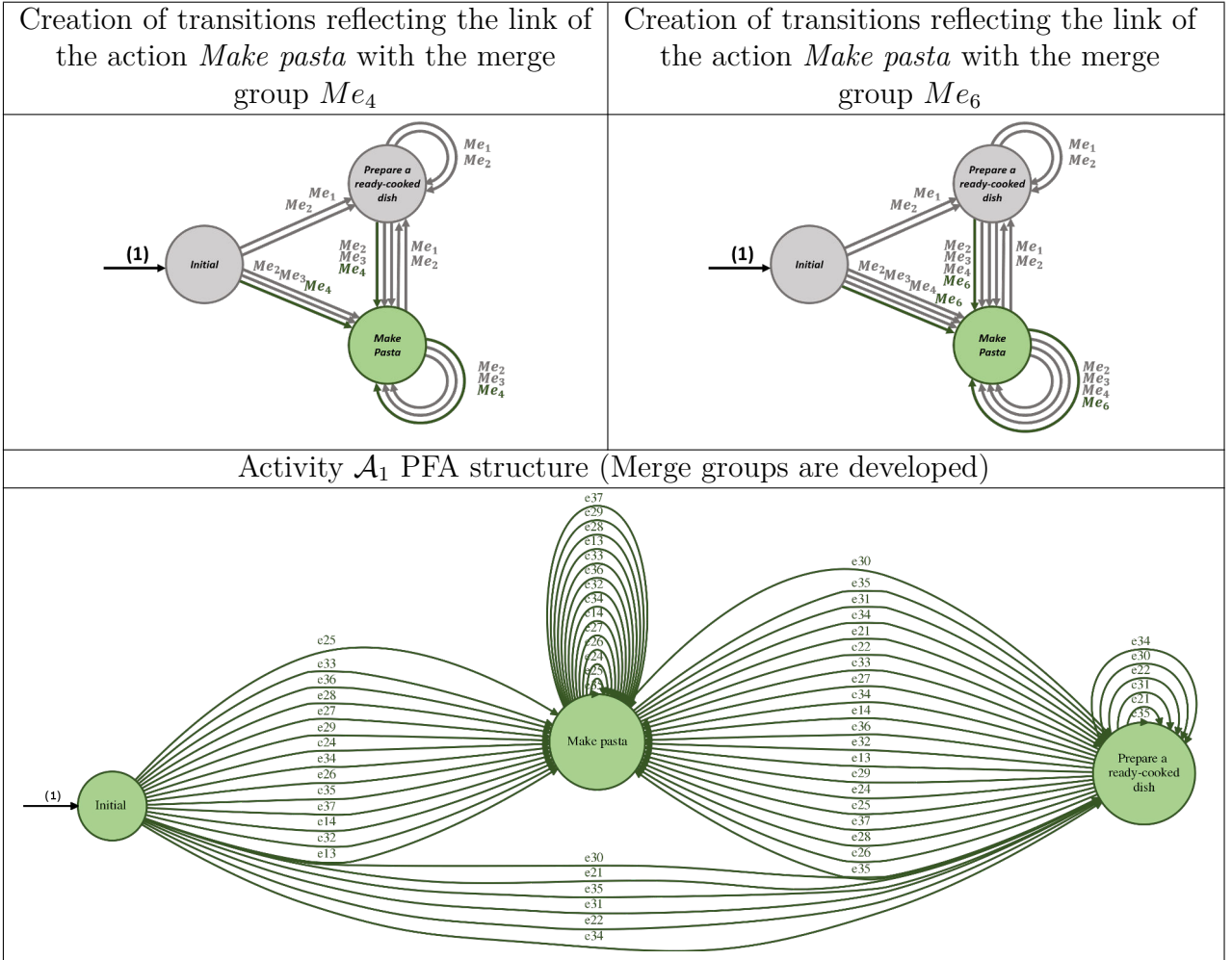


Figure 3.3.2: (a) Developed representation; (b) Merged representation to improve readability.

<p>States creation <math>Q_{\mathcal{A}_1} = \{Prepare\ a\ ready-cooked\ dish, Make\ pasta, Initial\}</math></p>	<p>Creation of transitions reflecting the link of the action <i>Prepare a ready-cooked dish</i> with the merge group <math>Me_1</math></p>
<p>Creation of transitions reflecting the link of the actions <i>Prepare a ready-cooked dish</i> and <i>Make pasta</i> with the merge group <math>Me_2</math></p>	<p>Creation of transitions reflecting the link of the action <i>Make pasta</i> with the merge group <math>Me_3</math></p>


 Figure 3.3.3: Generation of the activity  $\mathcal{A}_1$  (Cooking) PFA structure.

### 3.3.2 Database of event logs exploration

Once the structures of activities are known, we can compute probabilistic indicators linked to each activity  $\mathcal{A}_k$  presented in the previous section. These indicators will be extracted from the learning database generated as explained in Chapter 2.2.4. To do so, for the reasons explained before, the learning database is split in several sequences by sliding a window of a fix number of events. As the global learning database is composed of 1044 events and is therefore too long for being extensively treated, three short subsequence  $w_{\mathcal{A}_1}^{Obs}$ ,  $w_{\mathcal{A}_2}^{Obs}$  and  $w_{\mathcal{A}_3}^{Obs}$  of this global database will be used in this subsection to illustrate the indicators computation procedure. Each example subsequence represents a part of an occurrence of activities  $\mathcal{A}_1, \mathcal{A}_2$  and  $\mathcal{A}_3$ , respectively.

The chosen sequences used to illustrate the indicators computation are:

- $w_{\mathcal{A}_1}^{Obs} = e_{23} e_{14} e_{13} e_{37} e_{36} e_{23} e_{33} e_{32} e_{35}$
- $w_{\mathcal{A}_2}^{Obs} = e_{36} e_{29} e_{28} e_{23} e_{16} e_{15} e_{18} e_{17}$
- $w_{\mathcal{A}_3}^{Obs} = e_{39} e_{38} e_2 e_1 e_4 e_7 e_6 e_5 e_3$

Furthermore, those three retained sequences are representatives of three particular situations. Indeed:

- $w_{\mathcal{A}_1}^{Obs}$  is an activity  $\mathcal{A}_1$  performance of an activity sharing a lot of events with another one ( $w_{\mathcal{A}_2}^{Obs}$ ),



- $w_{\mathcal{A}_2}^{Obs}$  is an activity  $\mathcal{A}_2$  performance of an activity sharing a little part of events with an other one ( $w_{\mathcal{A}_1}^{Obs}$ ),
- $w_{\mathcal{A}_3}^{Obs}$  is an activity  $\mathcal{A}_3$  performance of an activity sharing no event with any other ones.

Table 3.3.1 shows the incrementation of the different counters during the analysis of the subsequences  $w_{\mathcal{A}_1}^{Obs}$ . Treatment of sequences  $w_{\mathcal{A}_2}^{Obs}$  and  $w_{\mathcal{A}_3}^{Obs}$  are given in appendix C. Values of the indicators  $N_{e_i}^k, N_{init\ e_i}^k$  and  $N_{e_i \rightarrow e_j}^k$  given in this case are initialised to zero at the beginning of the example treatment. Therefore, even if the analysed sequences are extracted from the case study database, the values taken by the counters are illustrative. The real values of our counters linked with the activity  $\mathcal{A}_1$  at the end of the learning period are given in the tables 3.3.2 to 3.3.4.

By analysing the count operation shown figures 3.3.1 and in annexe C, we can see than the same occurrence of an event is counted several time. For example, if not detected at the beginning or at the end of the learning database, the value  $N_{e_i}^k$  of an event  $e_i$  will be equal to the length of the sliding window multiply by the number of the event occurrences. Therefore, in figure 3.3.4, most of the events have a number  $N_{e_i}^k$  proportional to 5. This multiple count of the same occurrence is existing in the other indicators also. As the probabilities computation will consist in doing a ratio between all occurrences, this multiple count is not an issue.

At this step, all indicators are computed for each activity  $\mathcal{A}_k$  and we are now able to compute the transition probabilities.

Table 3.3.1: Counting example during the activity  $\mathcal{A}_1$  performance.

sequence $w_{\mathcal{A}_1}^{Obs} = e_{23} e_{14} e_{13} e_{37} e_{36} e_{23} e_{33} e_{32} e_{35}$				
sliding windows	projected sequences	$\mathcal{A}_1$ indicators changes	$\mathcal{A}_2$ indicators changes	$\mathcal{A}_3$ indicators changes
<i>Initial</i>	$\emptyset$	$N_{e_i}^1 = 0$ $\forall e_i \in \Sigma_{\mathcal{A}_1}$	$N_{e_i}^2 = 0$ $\forall e_i \in \Sigma_{\mathcal{A}_2}$	$N_{e_i}^3 = 0$ $\forall e_i \in \Sigma_{\mathcal{A}_3}$
$e_{23}e_{14}e_{13}e_{37}e_{36}$	$w_{p_1}^{Obs} : e_{14}e_{13}e_{37}e_{36}$ $w_{p_2}^{Obs} : e_{14}e_{13}e_{37}e_{36}$ $w_{p_3}^{Obs} : \emptyset$	$N_{init\ e_{14}}^1 \leftarrow 1$ $N_{e_{14}}^1 \leftarrow 1$ $N_{e_{13}}^1 \leftarrow 1$ $N_{e_{37}}^1 \leftarrow 1$ $N_{e_{36}}^1 \leftarrow 1$ $N_{e_{14} \rightarrow e_{13}}^1 \leftarrow 1$ $N_{e_{13} \rightarrow e_{37}}^1 \leftarrow 1$ $N_{e_{37} \rightarrow e_{36}}^1 \leftarrow 1$	$N_{init\ e_{14}}^2 \leftarrow 1$ $N_{e_{14}}^2 \leftarrow 1$ $N_{e_{13}}^2 \leftarrow 1$ $N_{e_{37}}^2 \leftarrow 1$ $N_{e_{36}}^2 \leftarrow 1$ $N_{e_{14} \rightarrow e_{13}}^2 \leftarrow 1$ $N_{e_{13} \rightarrow e_{37}}^2 \leftarrow 1$ $N_{e_{37} \rightarrow e_{36}}^2 \leftarrow 1$	$\emptyset$
$e_{14}e_{13}e_{37}e_{36}e_{23}$	$w_{p_1}^{Obs} : e_{14}e_{13}e_{37}e_{36}$ $w_{p_2}^{Obs} : e_{14}e_{13}e_{37}e_{36}$ $w_{p_3}^{Obs} : \emptyset$	$N_{init\ e_{14}}^1 \leftarrow 2$ $N_{e_{14}}^1 \leftarrow 2$ $N_{e_{13}}^1 \leftarrow 2$ $N_{e_{37}}^1 \leftarrow 2$ $N_{e_{36}}^1 \leftarrow 2$ $N_{e_{14} \rightarrow e_{13}}^1 \leftarrow 2$ $N_{e_{13} \rightarrow e_{37}}^1 \leftarrow 2$ $N_{e_{37} \rightarrow e_{36}}^1 \leftarrow 2$	$N_{init\ e_{14}}^2 \leftarrow 2$ $N_{e_{14}}^2 \leftarrow 2$ $N_{e_{13}}^2 \leftarrow 2$ $N_{e_{37}}^2 \leftarrow 2$ $N_{e_{36}}^2 \leftarrow 2$ $N_{e_{14} \rightarrow e_{13}}^2 \leftarrow 2$ $N_{e_{13} \rightarrow e_{37}}^2 \leftarrow 2$ $N_{e_{37} \rightarrow e_{36}}^2 \leftarrow 2$	$\emptyset$
$e_{13}e_{37}e_{36}e_{23}e_{33}$	$w_{p_1}^{Obs} : e_{13}e_{37}e_{36}e_{33}$ $w_{p_2}^{Obs} : e_{13}e_{37}e_{36}$ $w_{p_3}^{Obs} : \emptyset$	$N_{init\ e_{13}}^1 \leftarrow 1$ $N_{e_{13}}^1 \leftarrow 3$ $N_{e_{37}}^1 \leftarrow 3$ $N_{e_{36}}^1 \leftarrow 3$ $N_{e_{33}}^1 \leftarrow 1$ $N_{e_{13} \rightarrow e_{37}}^1 \leftarrow 3$ $N_{e_{37} \rightarrow e_{36}}^1 \leftarrow 3$ $N_{e_{36} \rightarrow e_{33}}^1 \leftarrow 1$	$N_{init\ e_{13}}^2 \leftarrow 1$ $N_{e_{13}}^2 \leftarrow 3$ $N_{e_{37}}^2 \leftarrow 3$ $N_{e_{36}}^2 \leftarrow 3$ $N_{e_{13} \rightarrow e_{37}}^2 \leftarrow 3$ $N_{e_{37} \rightarrow e_{36}}^2 \leftarrow 3$	$\emptyset$
$e_{37}e_{36}e_{23}e_{33}e_{32}$	$w_{p_1}^{Obs} : e_{37}e_{36}e_{33}e_{32}$ $w_{p_2}^{Obs} : e_{37}e_{36}$ $w_{p_3}^{Obs} : \emptyset$	$N_{init\ e_{37}}^1 \leftarrow 1$ $N_{e_{37}}^1 \leftarrow 4$ $N_{e_{36}}^1 \leftarrow 4$ $N_{e_{33}}^1 \leftarrow 2$ $N_{e_{32}}^1 \leftarrow 1$ $N_{e_{37} \rightarrow e_{36}}^1 \leftarrow 4$ $N_{e_{36} \rightarrow e_{33}}^1 \leftarrow 2$ $N_{e_{33} \rightarrow e_{32}}^1 \leftarrow 1$	$N_{init\ e_{37}}^2 \leftarrow 1$ $N_{e_{37}}^2 \leftarrow 4$ $N_{e_{36}}^2 \leftarrow 4$ $N_{e_{37} \rightarrow e_{36}}^2 \leftarrow 4$	$\emptyset$
$e_{36}e_{23}e_{33}e_{32}e_{35}$	$w_{p_1}^{Obs} : e_{36}e_{33}e_{32}e_{35}$ $w_{p_2}^{Obs} : e_{36}$ $w_{p_3}^{Obs} : \emptyset$	$N_{init\ e_{36}}^1 \leftarrow 1$ $N_{e_{36}}^1 \leftarrow 5$ $N_{e_{33}}^1 \leftarrow 3$ $N_{e_{32}}^1 \leftarrow 2$ $N_{e_{35}}^1 \leftarrow 1$ $N_{e_{36} \rightarrow e_{33}}^1 \leftarrow 3$ $N_{e_{33} \rightarrow e_{32}}^1 \leftarrow 2$ $N_{e_{32} \rightarrow e_{35}}^1 \leftarrow 1$	$N_{init\ e_{36}}^2 \leftarrow 1$ $N_{e_{36}}^2 \leftarrow 5$	$\emptyset$

Table 3.3.2:  $N_{e_i \rightarrow e_j}^1$  values for the activity  $\mathcal{A}_1$ , with  $e_i$  in column and  $e_j$  in line (e.g.  $N_{e_{13} \rightarrow e_{24}}^1 = 7$ ).

	$e_{13}$	$e_{14}$	$e_{21}$	$e_{22}$	$e_{24}$	$e_{25}$	$e_{26}$	$e_{27}$	$e_{28}$	$e_{29}$	$e_{30}$	$e_{31}$	$e_{32}$	$e_{33}$	$e_{34}$	$e_{35}$	$e_{36}$	$e_{37}$
$e_{13}$	0	0	0	0	7	4	0	0	0	0	0	0	0	0	0	6	0	65
$e_{14}$	86	0	0	0	4	0	4	4	0	0	0	0	0	0	0	0	0	0
$e_{21}$	0	0	0	0	0	0	0	0	0	0	0	36	0	0	0	4	0	0
$e_{22}$	0	0	40	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0
$e_{24}$	4	4	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0
$e_{25}$	0	0	0	0	4	0	0	0	4	0	0	0	0	15	0	0	0	0
$e_{26}$	3	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$e_{27}$	4	4	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0
$e_{28}$	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0
$e_{29}$	0	4	0	0	0	0	0	0	9	0	0	0	0	0	0	0	0	0
$e_{30}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	11	0	0
$e_{31}$	0	0	4	0	0	0	0	0	0	0	9	8	0	0	0	15	0	0
$e_{32}$	0	3	0	0	4	0	0	0	0	0	0	0	0	4	0	20	0	3
$e_{33}$	0	0	0	0	0	0	0	0	0	0	0	0	44	0	0	0	0	0
$e_{34}$	0	9	0	0	3	0	6	2	0	0	14	0	0	0	0	0	0	0
$e_{35}$	0	0	0	0	0	0	0	0	0	0	4	0	0	0	66	0	0	0
$e_{36}$	0	0	0	0	0	25	0	11	0	32	0	0	0	3	0	0	0	0
$e_{37}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	106	0

 Table 3.3.3:  $N_{init\ e_i}^1$  values for the activity  $\mathcal{A}_1$  (e.g.  $N_{init\ e_{13}}^1 = 35$ ).

$e_{13}$	$e_{14}$	$e_{21}$	$e_{22}$	$e_{24}$	$e_{25}$	$e_{26}$	$e_{27}$	$e_{28}$	$e_{29}$	$e_{30}$	$e_{31}$	$e_{32}$	$e_{33}$	$e_{34}$	$e_{35}$	$e_{36}$	$e_{37}$
35	99	11	52	18	11	10	8	32	13	23	12	11	27	20	29	44	81

 Table 3.3.4:  $N_{e_i}^1$  values for the activity  $\mathcal{A}_1$  (e.g.  $N_{e_{13}}^1 = 132$ ).

$e_{13}$	$e_{14}$	$e_{21}$	$e_{22}$	$e_{24}$	$e_{25}$	$e_{26}$	$e_{27}$	$e_{28}$	$e_{29}$	$e_{30}$	$e_{31}$	$e_{32}$	$e_{33}$	$e_{34}$	$e_{35}$	$e_{36}$	$e_{37}$
132	126	55	55	40	40	20	25	45	45	50	60	55	55	90	90	150	149

### 3.3.3 Probabilities computation

At this point, structures of activities  $\mathcal{A}_1$ ,  $\mathcal{A}_2$  and  $\mathcal{A}_3$  are established and needed indicators for the probabilities computation are known. Thus, all needed information is in our hands to compute our model probabilities.

In order to illustrate this computation step, application of equations (3.8) to (3.10) for some transitions of the activity  $\mathcal{A}_1$  are now presented. These application examples use indicator values shown tables 3.3.2 to 3.3.4.

**Application to the transition (*Initial*,  $e_{13}$ , *Make Pasta*) (case  $q_g = q_0$  of equation (3.8)):**

According to equation (3.10), we have:

$$P(\text{Initial}, e_{13}, \text{Make Pasta}) = P(\text{Initial} \rightarrow \text{Make Pasta} | \text{Initial}) \times P(e_{13} | \text{Initial} \rightarrow \text{Make Pasta})$$

with:

$$P(\text{Initial} \rightarrow \text{Make Pasta} | \text{Initial}) = \frac{\tilde{N}(\text{Initial} \rightarrow \text{Make Pasta} | \text{Initial})}{\sum_{q_n \in Q_{\mathcal{A}_1}} \tilde{N}(\text{Initial} \rightarrow q_n | \text{Initial})} \quad (3.16)$$

$$P(e_{13} | \text{Initial} \rightarrow \text{Make Pasta}) = \frac{\tilde{N}(e_{13} | \text{Initial} \rightarrow \text{Make Pasta})}{\sum_{e_j \in \Sigma_{\mathcal{A}_1}} \tilde{N}(e_j | \text{Initial} \rightarrow \text{Make Pasta})}$$

According to equation (3.8):

$$\begin{aligned} \tilde{N}(\text{Initial} \rightarrow \text{Make Pasta} | \text{Initial}) &= \sum_{e_i \in \Sigma_{\text{Make Pasta}}} \frac{1}{C_{e_i}} N_{init\ e_i}^1 \\ &= \begin{array}{cccccc} N_{init\ e_{13}}^1 & + & N_{init\ e_{14}}^1 & + & N_{init\ e_{24}}^1 & + & N_{init\ e_{25}}^1 & + \\ N_{init\ e_{26}}^1 & + & N_{init\ e_{27}}^1 & + & N_{init\ e_{28}}^1 & + & N_{init\ e_{29}}^1 & + \\ N_{init\ e_{32}}^1 & + & N_{init\ e_{33}}^1 & + & \frac{1}{2} N_{init\ e_{34}}^1 & + & \frac{1}{2} N_{init\ e_{35}}^1 & + \\ N_{init\ e_{36}}^1 & + & N_{init\ e_{37}}^1 & & & & & \end{array} \\ &= \begin{array}{cccccc} 35 & + & 99 & + & 18 & + & 11 & + \\ 10 & + & 8 & + & 32 & + & 13 & + \\ 11 & + & 27 & + & 20/2 & + & 29/2 & + \\ 44 & + & 81 & & & & & \end{array} \\ &= 413.5 \end{aligned} \quad (3.17)$$

Analogously, we can find:

$$\tilde{N}(\text{Initial} \rightarrow \text{Prepare a ready cooked dish} | \text{Initial}) = 122.5 \quad (3.18)$$

thus:

$$P(\text{Initial} \rightarrow \text{Make Pasta} | \text{Initial}) = \frac{413.5}{413.5 + 122.5} \simeq 0.7714 \quad (3.19)$$

Then, according to equation (3.9):

$$\begin{aligned}
 \tilde{N}(e_{13}|Initial \rightarrow Make\ Pasta) &= N_{e_{13}}^1 = 132 \\
 \text{and} \\
 \sum_{e_j \in \Sigma_{A_1}} \tilde{N}(e_j|Initial \rightarrow Make\ Pasta) &= 132 + 126 + 40 + 40 + 20 + 25 + 45 \\
 &\quad + 45 + 55 + 55 + 90 + 90 + 150 + 149 \\
 &= 1062
 \end{aligned} \tag{3.20}$$

thus:

$$P(e_{13}|Initial \rightarrow Make\ Pasta) = \frac{132}{1062} \simeq 0.1243 \tag{3.21}$$

Finally, the probability  $P(Initial, e_{13}, Make\ Pasta)$  is:

$$P(Initial, e_{13}, Make\ Pasta) = \frac{413.5}{413.5 + 122.5} \times \frac{132}{1062} \simeq 0.0959 \tag{3.22}$$

**Application to the transition ( $Make\ Pasta, e_{13}, Make\ Pasta$ ) (joining cases  $q_g = q_h$  and  $q_g \neq q_h \neq q_0$  of equation (3.8)):**

According to equation (3.10), we have:

$P(Make\ Pasta, e_{13}, Make\ Pasta) = P(Make\ Pasta \rightarrow Make\ Pasta|Make\ Pasta) \times P(e_{13}|Initial \rightarrow Make\ Pasta)$   
with:

$$\begin{aligned}
 P(Make\ Pasta \rightarrow Make\ Pasta|Make\ Pasta) &= \frac{\tilde{N}(Make\ Pasta \rightarrow Make\ Pasta|Make\ Pasta)}{\sum_{q_n \in Q_{A_1}} \tilde{N}(Make\ Pasta \rightarrow q_n|Make\ Pasta)} \\
 P(e_{13}|Make\ Pasta \rightarrow Make\ Pasta) &= \frac{\tilde{N}(e_{13}|Make\ Pasta \rightarrow Make\ Pasta)}{\sum_{e_j \in \Sigma_{A_1}} \tilde{N}(e_j|Make\ Pasta \rightarrow Make\ Pasta)}
 \end{aligned} \tag{3.23}$$

According to equation 3.8:

$$\begin{aligned}
 \tilde{N}(Make\ Pasta \rightarrow Make\ Pasta|Make\ Pasta) &= \sum_{e_i, e_j \in \Sigma_{Make\ Pasta}} N_{e_i \rightarrow e_j}^1 \\
 &= N_{e_{13} \rightarrow e_{13}}^1 + N_{e_{13} \rightarrow e_{14}}^1 + N_{e_{13} \rightarrow e_{24}}^1 + N_{e_{13} \rightarrow e_{25}}^1 + N_{e_{13} \rightarrow e_{26}}^1 + N_{e_{13} \rightarrow e_{27}}^1 + \dots \\
 &= 590
 \end{aligned} \tag{3.24}$$

and

$$\begin{aligned}
 \tilde{N}(Make\ Pasta \rightarrow Prepare\ a\ ready\ cooked\ dish|Make\ Pasta) &= \\
 &= \sum_{\substack{e_i \in \Sigma_{Make\ Pasta} \\ e_j \in \Sigma_{Prepare\ a\ ready\ cooked\ dish}}} \frac{1}{C_{e_j}} N_{e_i \rightarrow e_j}^1 \\
 &= N_{e_{13} \rightarrow e_{21}}^1 + N_{e_{13} \rightarrow e_{22}}^1 + N_{e_{13} \rightarrow e_{30}}^1 + N_{e_{13} \rightarrow e_{31}}^1 + N_{e_{14} \rightarrow e_{21}}^1 + N_{e_{14} \rightarrow e_{22}}^1 + \dots \\
 &= 21
 \end{aligned} \tag{3.25}$$

thus:

$$P(Make\ Pasta \rightarrow Make\ Pasta|Initial) = \frac{590}{590 + 21} \simeq 0.9656 \tag{3.26}$$

Concerning the probability for the event to occur, according to equation 3.9,  $\tilde{N}(e_i|q_g \rightarrow q_h)$  does not depend on  $q_g$ :  $P(e_i|q_g \rightarrow q_h) = P(e_i|q'_g \rightarrow q_h) \forall q_g, q'_g \in Q_{A_k}$ . Its give us:

$$P(e_{13}|Make\ Pasta \rightarrow Make\ Pasta) = P(e_{13}|Initial \rightarrow Make\ Pasta) = \frac{132}{1062} \simeq 0.1243 \quad (3.27)$$

Finally, the probability  $P(Make\ Pasta, e_{13}, Make\ Pasta)$  is as follows:

$$P(Make\ Pasta, e_{13}, Make\ Pasta) = \frac{590}{590 + 21} \times \frac{132}{1062} \simeq 0.1200 \quad (3.28)$$

By automatically apply those methods to all transitions, we obtain the complete activity  $\mathcal{A}_1$ ,  $\mathcal{A}_2$  and  $\mathcal{A}_3$  models presented in figures 3.3.4, 3.3.5 and 3.3.6, respectively. Those models will be discussed in the next section and they will be used as input models of the example in the next chapter.

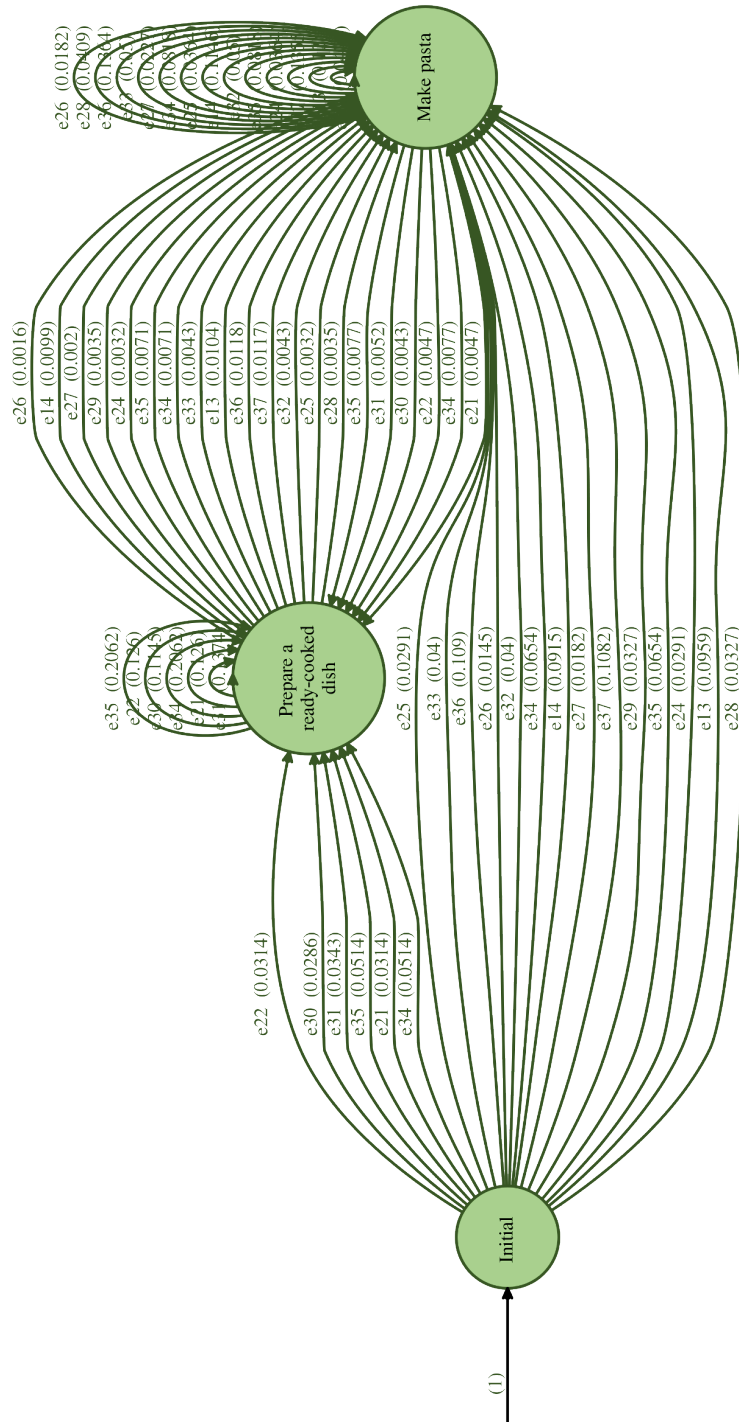


Figure 3.3.4: Final  $\mathcal{A}_1$  *Cooking* model.

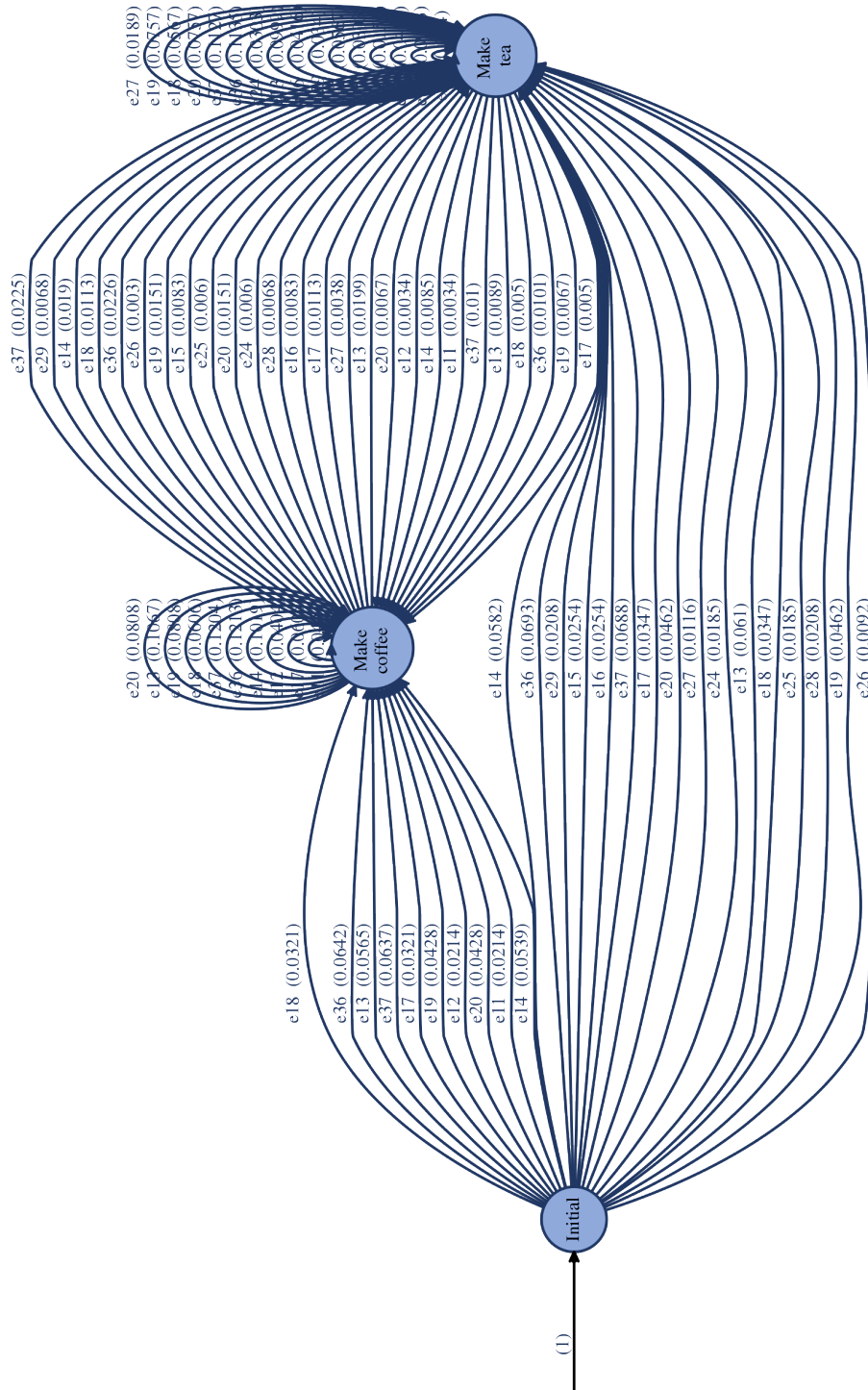


Figure 3.3.5: Final  $A_2$  Hot beverage preparation model.



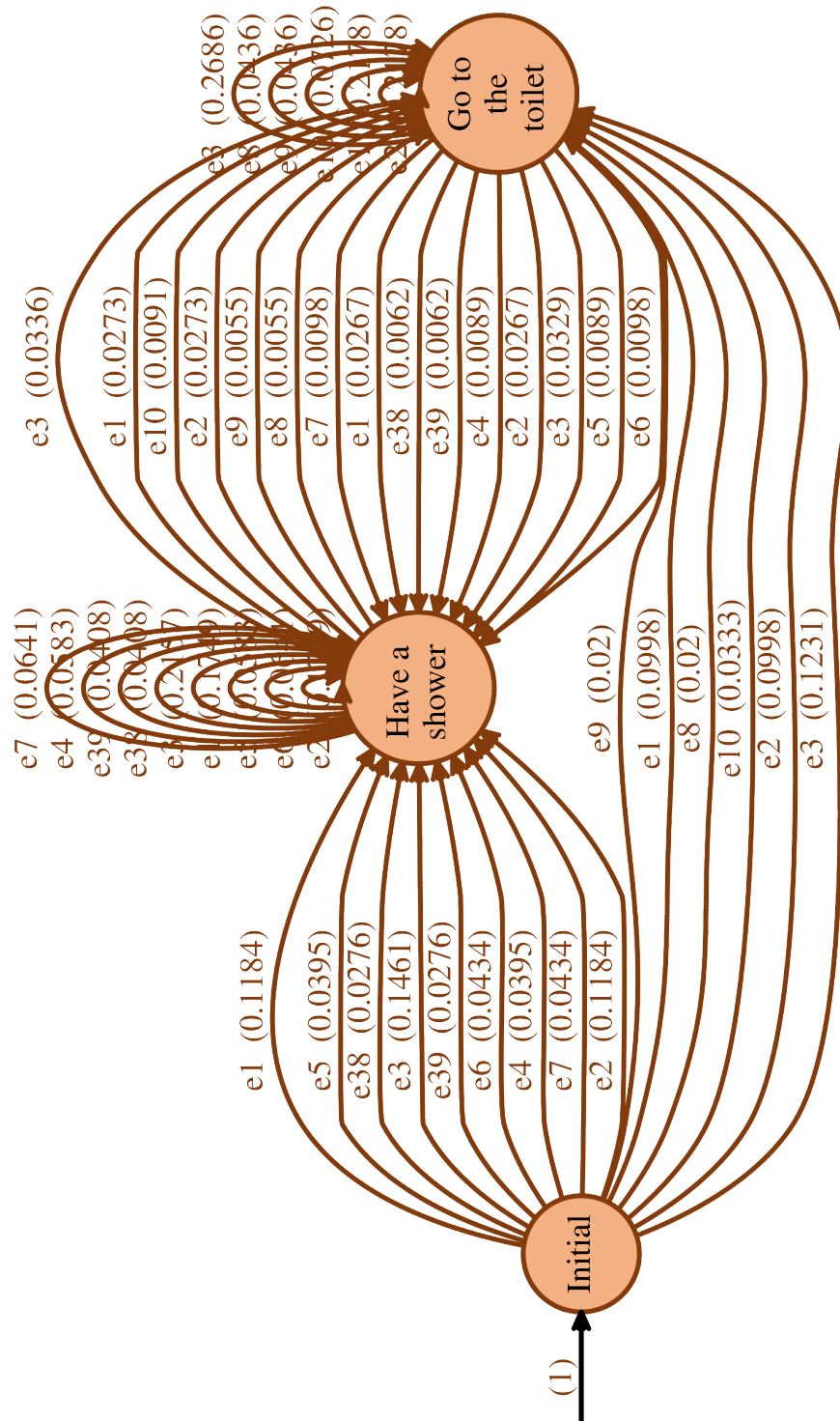


Figure 3.3.6: Final  $\mathcal{A}_3$  *Use bathroom* model.

### 3.4 Discussion

By analysing the model of the activity  $\mathcal{A}_1$  shown in figure 3.3.4, we can observe that the most observed event  $e_{36}$  (observed 150 times in a sequence of 1044 events) of this activity is the one with the highest probability in the model for each couple of actions in which it is implied. This result was expected since, for a fixed state transition  $q_g \rightarrow q_h$ , the probability  $P(q_g, e_i, q_h)$  is directly proportional to the counter  $N_{e_i}^k$  (equations (3.10) and (3.9)).

This logic, expected and wanted property can be a disadvantage since, if shared between several activities, it is impossible to distinguish occurrences of its events is arriving with one activity or with another one. The most observed events  $e_{36}$  and  $e_{37}$  of the activity  $\mathcal{A}_1$  are submitted to this skewed since their are also linked to the activity  $\mathcal{A}_2$ , especially with the action *make tea* of  $\mathcal{A}_2$ .

In the presented method, we consider important to have this drawback in mind during the use of the models. This drawback is unavoidable without to record the activity performed by the inhabitant during the learning period. However, the main advantage of this method is precisely not to use this knowledge (Assumption 4), systematically difficult to obtain and not reliable. The pointed out drawback is negligible in comparison with the use of a potentially faulty input.

### Conclusion

In this chapter, an approach for activity of daily living discovery was proposed and illustrated. A procedure to model activities by probabilistic finite-state automata was developed based on the knowledge of a training event logs database and the decomposition of activities into actions expertly linked with the smart home events. The main advantage of this method is related to the Assumption 4 of this thesis avoiding the use of activity labelling during the learning phase to generate the models.



## Activity Recognition

---

*"Never ignore coincidence. Unless, of course, you're busy.  
In which case, always ignore coincidence."*

- The 11th Doctor -

Steven Moffat, *Doctor Who, Season 5, Episode 12: The Pandorica Opens*

### Abstract

---

This chapter proposes an approach allowing to recognise in real time which activity is being performed by an inhabitant. For that, a new probabilistic distance is proposed and the robustness of this indicator as well as its computational complexity are discussed. Finally, the quality of the obtained results are shown on the basis of experiments performed in the experimental smart flat.

---

### Contents

---

Introduction . . . . .	76
4.1 Models and used notations . . . . .	76
4.2 Activity Recognition protocol . . . . .	77
4.2.1 Observed sequence windowing . . . . .	77
4.2.2 Projection of the exploited sequence . . . . .	78
4.2.3 Language creation using the projected sequence . . . . .	79
4.2.4 Probability estimation . . . . .	81
4.3 Distance between a sequence and a PFA . . . . .	82
4.3.1 The likelihood . . . . .	82
4.3.2 The perplexity . . . . .	83
4.3.3 Issues due to distance hypotheses . . . . .	84
4.4 Normalised likelihood and extension . . . . .	85
4.4.1 The normalised likelihood . . . . .	85
4.4.2 The normalised perplexity . . . . .	86
4.4.3 Computational Issues . . . . .	87
4.5 Application to the Case Study . . . . .	96
4.5.1 Observed sequence windowing . . . . .	96
4.5.2 Projection of the exploited sequence . . . . .	96
4.5.3 Language generation based on the projected sequence . . . . .	97
4.5.4 Probability estimation . . . . .	100
4.6 Discussion . . . . .	102
4.6.1 Activities occurring individually . . . . .	102
4.6.2 Activities occurring successively . . . . .	104
4.6.3 Validation . . . . .	105
Conclusion . . . . .	105

---

## Introduction

In this chapter, the second main contribution of this thesis is presented: a new activity recognition (AR) method. Activity recognition consists in determining on-line, potentially in real-time if required, the activity performed by a smart home inhabitant during its operation. To this aim, the group {inhabitant + smart home} is considered as an event generator and the activities to monitor are modelled by PFAs. As explained in section 4.3, in the literature, it exists several methods to recognise which PFA, among several, generates a given sequence if the compared PFAs have the same alphabet  $\Sigma_{A_k}$ . Otherwise, if PFAs do not share the same alphabet, existing methods are not applicable. Unfortunately, our models generated in chapter 3, are linked to several activities of daily living and have different alphabets. Therefore, the new AR method presented in this chapter is needed.

A common point between the existing methods (Van Kasteren et al., 2008; Kellokumpu et al., 2005) and ours is the definition and the use of a *distance*. Indeed, in all methods, a distance computation is used to estimate which activity is the most probably performed.

Furthermore, in order to improve the recognition, several operations, as projections and language computation, have to be considered. The projection helps to avoid noise by keeping only useful events for each activity as explained in section 4.2.2. The observed language generation allows considering the observed sequence of events as a succession of possibly independent event groups, as described in section 4.2.3.

To put it in a nutshell, after a brief definition of languages and some complements about the sequences terminology, made in section 4.1, the proposed protocol to recognise an activity among several modelled by PFAS is detailed in the section 4.2. The last step of this protocol (i.e. the distance computation) requires the definition of a new indicator. Therefore, already existing distances between a sequence and an automata will be presented in section 4.3. Limits of these distances will be enlightened in regard to their use with our models and the observed languages. Thus, an original distance indicator compatible with our models is presented in section 4.4 and computation issues linked to this distance are also treated. Finally, an application to case study of the chapters 2 and 3 will be presented in section 4.5 and discussed in section 4.6.

### 4.1 Models and used notations

In this section, in addition to the definition of PFAs and the implied concepts given in section 3.1, the definition of a language is given here. In addition, some specific notations needed in this chapter are presented.

**Definition 4.1** (Language (Cassandras and Lafortune, 2009)). *A language defined over an event set  $\Sigma$  is a set of finite-length sequence (or strings) formed from events in  $\Sigma$ .*

A language over an event set  $\Sigma$  is therefore a subset of  $\Sigma^*$  which is the Kleene-closure of the set of events  $\Sigma$ , i.e. the infinite set of all possible sequences of elements of  $\Sigma$ . In particular,  $\emptyset$ ,  $\Sigma$ , and  $\Sigma^*$  are languages.

In order to well define languages, some notions and definitions about sequences have to be known. Therefore, sequence prefix, suffix and substring are defined as the following:

**Definition 4.2** (prefix, suffix and substring). *Let  $tuv = s$  be a sequence in  $\Sigma^*$  with  $t, u, v \in \Sigma^*$ , then:*

- $t$  is called a prefix of  $s$ ,
- $u$  is called a substring of  $s$ , and
- $v$  is called a suffix of  $s$ .

We observe that both  $\epsilon$  and  $s$  are prefixes, substrings and suffixes of  $s$ .

In addition, we choose to denote  $\Sigma_{\mathcal{A}_k}^m$  the language composed of all possible sequences of length  $m$  composed with events of  $\Sigma_{\mathcal{A}_k}$ .

## 4.2 Activity Recognition protocol

In order to recognise on-line the performed activity using the observed events and the previously built models, we propose a four-step protocol. The four steps are graphically represented in figure 4.2.1 and are the following:

- ① The observed event sequence  $e_i \dots e_j$  is split in order to treat the closed events: the observed sequence  $w^{Obs}$  is thus obtained,
- ② The obtained sequence is projected to remove disturbances: the projected sequence  $w_{p_k}^{Obs}$  is resulting,
- ③ A language  $\mathcal{L}_{p_k}^{Obs}$  is created using the projected sequence to translate the continuous nature of the observation,
- ④ Finally, occurrence probability of each activity is computed using a specific distance indicator.

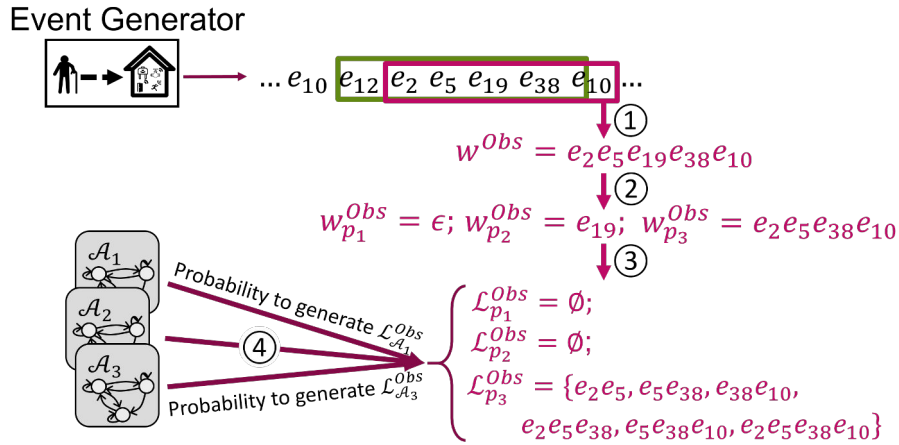


Figure 4.2.1: The proposed activity recognition protocol

This protocol is the result of several reflexions and tests. Each of those steps is essential to have a good recognition of activities and the necessity of each one and their operations are presented below.

### 4.2.1 Observed sequence windowing

As an inhabitant living at home is considered as an event generator on which we have no influence, the recognition of the performed activity has to be done by using only the generated events and the activity models at our disposition. As the inhabitant

at home generates new events continuously, the objective of this step is to choose the sequence in which the recognition is performed.

In order to recognise activity on-line, it is necessary to perform an efficient and quick activity recognition each time a new event is observed. As an activity is a complex succession of events, the recognition cannot be made by only using the last observed event: an appropriate view of the past is needed. Therefore, during the activity recognition, a sequence of events has to be considered. This sequence consists in the last occurred event and some of the previous ones. Concerning the length of the retained sequence, a trade-off has to be done since the kept sequence of events should be long enough to traduce the current human behaviour but not too much to have an acceptable computation time and being representative of the activity to be recognised. The *sliding* operation in this case is not a window that we change to scan a learning database, but the list of the last observed events evolving since the time is running.

As for the windowing performed in section 3.2.2, the three most common approaches are:

- (i) explicit segmentation;
- (ii) time base windowing;
- (iii) sensor event based windowing.

As precise in section 3.2.2, the sensor event based windowing is the most adapted method to treat activities with variable duration and separated with a variable duration.

The main difference with the sensor windowing performed in chapter 3 is the constraint around the window length. Indeed, in chapter 3, the window length has mainly to be chosen short enough to reduce the computational time but not too short to avoid the generation of too much sequences from the learning database. However, in this chapter, the sequences must have, as much as possible, the length of the performed activity.

Unfortunately, there is no existing systematic method to compute the windows length *a priori*.

However, by experiment, we observe that the optimal size of the sliding windows (i.e. the size giving the bests AR results) is strongly linked to the noise during the life observation. Is considered as *noisy* each event occurring during an activity occurrence and not hierarchically linked with it. More noise there is, more the sequence has to be long.

Once the sequence is extracted from on-line flow, we have to deal with noises.

### 4.2.2 Projection of the exploited sequence

In the normal life, sensors of smart homes useful to recognise activities of daily living are subjected to several kind of disturbance. First of all, some sensors of the smart home useful to other missions (inactivity detection, motion sensors used for the inhabitant security, ...) can flood the recorded database. Secondly, some sensors could malfunction and send events in excess. Finally, the human indeterminism can lead the monitor person to perform useless move or to stroll in the home.

Furthermore, sensors implanted in the smart home to detect a specific activity  $\mathcal{A}_{k1}$  but emitting when another activity  $\mathcal{A}_{k2}$  is performed can be considered as a noise for this specific activity  $\mathcal{A}_{k2}$ . This situation can occur if the two activities  $\mathcal{A}_{k1}$  and  $\mathcal{A}_{k2}$  share

the same geographic area. Thus, the projection allows recognising activities performed in a same area but implying different sensors locally.

Our objective is to use probabilistic distances to detect the performed activity. Unfortunately, using these kinds of distance, if a totally unexpected event occurs, the obtain result diverges quickly. This characteristic, explained later in section 4.3, leads to envisage a projection of the observed sequence  $w^{Obs}$  to each model alphabet  $\Sigma_{\mathcal{A}_k}$ , as defined in section 3.1, before computing distance between the sequence  $w^{Obs}$  and the PFA  $\mathcal{A}_k$ . Thus, the sequences  $w_{p_k}^{Obs}$  used to estimate the probability for each activity  $\mathcal{A}_k$  to be performed are shorter than the observed sequence but contain only non-noisy events.

By performing the projection of the observed sequence in each PFA alphabet, we obtain a different projected window for each activity. This point is appreciable since, in this way, the probability for each activity to be detected as occurring will not depend on the others. Furthermore, an activity semantically more general than another one will not flow up another activity generating a very low number of events. In brief, the projection act as a filter.

We remark that, if the inhabitant performs an activity  $\mathcal{A}_k$  sharing no events with any other activity, it is highly likely that the projected sequences linked to the other activities are empty.

The influence of the length of the treated window chosen in the previous step is noted at this step. On the one hand, if the length of the sequence is too small compared to the number of noisy events, a very small number of events are kept at the end and the future results will not be significant. On the other hand, if the chosen length is too long, too much events will be kept after the projection step since most of the manipulated sequences could traduce several activities generating a small number of events.

### 4.2.3 Language creation using the projected sequence

The life performance is a continuous succession of actions and activities. Transitions between these activities are most of the time non-instantaneous but smooth. As a consequence, observed events traduce this life property. By choosing to treat a continuously sliding window of a fix length to perform the AR computation, we are compatible with it since, if activities are changed, new events arrive one by one in the considered sequence.

However, it is necessary not to consider the sequences  $w_{p_k}^{Obs}$  obtained after projection as a block. Indeed, in a long sequence, several phenomenon can be included and a local succession of events can have a higher importance than the global sequence. Thus, in the same way as the sliding window performed step 1 traduce a continuity in the monitored person life, a smaller sliding window in the obtained projected sequence can also be important. For example, let's imagine a sequence  $w_{p_k}^{Obs} = e_1e_2e_4e_6$  where the succession of events  $e_2e_4$  is a representative behaviour of a given activity  $\mathcal{A}_k$ , and the two other events ( $e_1$  and  $e_6$ ) are linked to  $\mathcal{A}_k$  but not as much than the two others from a probabilistic point of view. In this case, it can be better to read the sequence as the successions of events  $e_1e_2$ ,  $e_2e_4$  and  $e_4e_6$ . In this way, the representative succession of events can be treated independently, as well as the two non-representative ones. The obtained result can thus procure a probabilistic result closest with the realty than by considering just the global sequence  $e_1e_2e_4e_6$ . Thus, in the presented AR method, every succession of events can have an importance.

Therefore, in order not to consider the last observed events as a block, a language is created. Several choices to generate this language exists. The most common ([Cassandras](#)



and Lafortune, 2009) are the following:

**The prefix-closed language:** Let  $w_{p_k}^{Obs} \in \Sigma_{\mathcal{A}_k}^*$ , then

$$\mathcal{L}_{pref}(w_{p_k}^{Obs}) = \{s \in \Sigma_{\mathcal{A}_k}^* \mid \exists t \in (\Sigma_{\mathcal{A}_k} \cup \{\epsilon\})^* \text{ such as } st = w_{p_k}^{Obs}\}. \quad (4.1)$$

In words, the language prefix-close of the sequence  $w_{p_k}^{Obs}$  is the language consisting of all the prefixes of  $w_{p_k}^{Obs}$ .

In our work, this language traduces the time running starting from an event of reference. It can be useful if we want to study the consequences linked to the observation of a given event.

**The suffix-closed language:** Let  $w_{p_k}^{Obs} \in \Sigma_{\mathcal{A}_k}^*$ , then

$$\mathcal{L}_{suff}(w_{p_k}^{Obs}) = \{s \in \Sigma_{\mathcal{A}_k}^* \mid \exists t \in (\Sigma_{\mathcal{A}_k} \cup \{\epsilon\})^* \text{ such as } ts = w_{p_k}^{Obs}\}. \quad (4.2)$$

In words, the language suffix-close of the sequence  $w_{p_k}^{Obs}$  is the language consisting of all the suffixes of  $w_{p_k}^{Obs}$ .

In our work, this language traduces successions of events that may have led to a given event (the last one). It can be useful if we want to study the reasons for a specific situation.

**Language of substrings:** Let  $w_{p_k}^{Obs} \in \Sigma_{\mathcal{A}_k}^*$ , then

$$\mathcal{L}_{<w_{p_k}^{Obs}} = \sum_{n=1}^{|w_{p_k}^{Obs}|} \mathcal{L}_{w_{p_k}^{Obs}}^n; \quad (4.3)$$

with:

$$\mathcal{L}_{w_{p_k}^{Obs}}^n = \{w \in w_{p_k}^{Obs} \mid \exists s, t \in (\Sigma_{\mathcal{A}_k} \cup \{\epsilon\})^* \text{ such as } swt = w_{p_k}^{Obs} \text{ and } |w| = n\} \quad (4.4)$$

In words, the language substrings of the sequence  $w_{p_k}^{Obs}$  is the language consisting of all the substrings of  $w_{p_k}^{Obs}$ .

In our work, this language traduces each succession of events observed in our sequence (that includes their prefixes and suffixes). It is useful if we want to study the reasons and the consequences of all observed events, and it is our case. Indeed, as we precise before, each succession of events has an importance. Moreover, since each activity can be interrupted at any time, all words included in the sequence have to be kept.

Since it is impossible by analysing an isolated event to conclude if it represents an activity realisation or a noisy phenomenon, all substrings of length one are excluded from the language used in our method. Thus, the language actually used is the following:

$$\mathcal{L}_{2 < w < w_{p_k}^{Obs}} = \sum_{n=2}^{|w_{p_k}^{Obs}|} \mathcal{L}_{w_{p_k}^{Obs}}^n \text{ with } \mathcal{L}_{w_{p_k}^{Obs}} = \{w \in w_{p_k}^{Obs}, |w| = n\}. \quad (4.5)$$

and can also be written as:

$$\mathcal{L}_{2 < w < w_{p_k}^{Obs}} = \sum_{n=2}^{|w_{p_k}^{Obs}|} \{\text{all possible substrings of } w_{p_k}^{Obs} \text{ of length } l\}. \quad (4.6)$$

Thereafter, the language  $\mathcal{L}_{2 < w < w_{p_k}^{Obs}}$  will be denoted  $\mathcal{L}_{p_k}^{Obs}$ . An example of this language generation is visible figure 4.2.2.

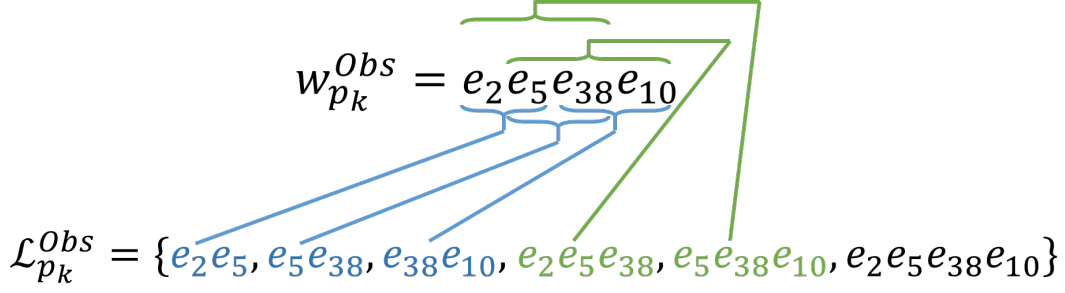


Figure 4.2.2: Chosen language generation: language of substrings of length upper than two.

The language representing the human behaviour during the closed past is now generated. The following step treats this language to obtain a high level information: the probability for each activity  $\mathcal{A}_k$  to be performed.

#### 4.2.4 Probability estimation

As first inputs of this step, we have several PFAs modelling several activities  $\{\mathcal{A}_k\}$ . These models are obtained by applying the AD method presented in chapter 3 in a database recorded during a learning period of the smart home inhabitant life. Activity models are never modified during the recognition. Secondly, the observed language  $\mathcal{L}_{p_k}^{Obs}$  generated from the observed sequence  $w^{Obs}$  as explained in section 4.2.3 are needed. The generation of these languages is chosen to represent the fragmented but continuous property of the human life.

Starting from these data, the objective of this final step is to estimate the probability for each activity model  $\mathcal{A}_k$  to generate the corresponding language  $\mathcal{L}_{p_k}^{Obs}$ . This probability, denoted  $P(\mathcal{L}_{p_k}^{Obs} | \mathcal{A}_k)$ , is also called the distance between the language  $\mathcal{L}_{p_k}^{Obs}$  and the PFA  $\mathcal{A}_k$ .

The algorithm 4 gives a pseudo-code representation of the presented protocol. As the used distance is presented in the next subsections, its computation is not detailed yet. The complexity  $C_{AR}$  of the AR protocol is polynomial considering the length of the observed sequence  $w^{Obs}$  and linear considering the distance complexity:

$$C_{AR} = O\left(\text{card}(\text{ActivityPFAs}) \times (|w^{Obs}|^2 + C_{distance})\right). \quad (4.7)$$

with:

- $\text{card}(\text{ActivityPFAs})$  the number of activities to monitor;
- $|w^{Obs}|$  the length of the observed sequence;
- $C_{distance}$  the complexity of the retained distance.

Several probability distances exist but the most used to compute the distance between a language and a probability model such as PFA are the *likelihood* based methods (Rabiner, 1989) such as the *perplexity* (Vidal et al., 2005b). The *likelihood* is a distance between a sequence and a PFA and is used as a base to compute the *perplexity*. Those two distances are presented in subsection 4.3.

---

**Algorithm 4** AR protocol. The distance computation is not detailed.

---

**Require:**

- A set of PFAs modelling the activities to monitor:  
 $ActivityPFAs : \{\mathcal{A}_1 \dots \mathcal{A}_k\}$ ,  
with  $\mathcal{A}_k = \langle Q_{\mathcal{A}_k}, \Sigma_{\mathcal{A}_k}, \delta_{\mathcal{A}_k}, I_{\mathcal{A}_k}, F_{\mathcal{A}_k}, P_{\mathcal{A}_k} \rangle$
- An event sequence  $w^{Obs}$  of the last observed events. //step 1: observed sequence windowing.

**Ensure:** Distances between the given activities and the observed sequence  $w^{Obs}$ .

```

1: // Global Initialisation:
2: dictDistances :=  $\emptyset$  // Dictionary of distances.
3: // For all activity models (and by extension, for all activities):
4: for all  $\langle Q_{\mathcal{A}_k}, \Sigma_{\mathcal{A}_k}, \delta_{\mathcal{A}_k}, I_{\mathcal{A}_k}, F_{\mathcal{A}_k}, P_{\mathcal{A}_k} \rangle \in ActivityPFAs$  do
5:   // Step 2: Projection of the exploited sequence:
6:    $w_{p_k}^{Obs} := Proj(w^{Obs}, \Sigma_{\mathcal{A}_k})$ 
7:   // Step 3: Language creation using the projected sequence:
8:   language :=  $\emptyset$ 
9:   lenSubStr := 2 // Substring of length 2 are first selected:
10:  // Selection of the substrings:
11:  while lenSubStr  $\leq |w_{p_k}^{Obs}|$  do
12:    for index_subStr := 1 to  $|w_{p_k}^{Obs}| - lenSubStr + 1$  do
13:      append  $w_{p_k}^{Obs}[index\_subStr \text{ to } index\_subStr + lenSubStr - 1]$  to language
14:    end for
15:    lenSubStr := lenSubStr + 1
16:  end while
17:  // Step 4: Probability estimation:
18:  distance :=  $P(\mathcal{L}_{p_k}^{Obs} | \mathcal{A}_k)$ 
19:  dictDistances[ $\mathcal{A}_k$ ] := distance
20: end for

```

---

As explained in subsection 4.3.3, these two presented distances are not adapted in our case since the sequence lengths of the languages to compare are, in the majority of cases, different. Therefore, in our work, we extend the existing likelihood then the perplexity to deal with this issue. The original distances thus obtained, called *normalised likelihood* and *normalised perplexity*, are presented in subsection 4.4.

It is important to note here that the proposed method is modular. The AR can be performed on several activities having their own models. If a new activity has to be recognised, we just have to add the new model. Furthermore, if we want to apply a new distance method to the observed events or to use a new language generation process, it is not necessary to change all the procedure, but only the corresponding step.

## 4.3 Distance between a sequence and a PFA

### 4.3.1 The likelihood

The *likelihood*, as defined in Vidal et al. (2005a) and Rabiner (1989), is the classical distance, denoted  $P(w^{Obs} | \mathcal{A}_k)$ , between a sequence  $w^{Obs}$  and a PFA  $\mathcal{A}_k$ . By computing this distance, an estimation of the probability that the PFA  $\mathcal{A}_k$  generates a sequence  $w^{Obs} \in \Sigma_{\mathcal{A}_k}^*$  is done.

To deal with this probability computation, let  $\theta = (q_l, w'_1, q_m, w'_2, q_s, \dots, q_p, w'_{|w^{Obs}|}, q_r)$

be a path of transitions for  $w^{Obs}$  in  $\mathcal{A}_k$ ; it signifies that there is a sequence of transitions  $(q_l, e_i, q_m), (q_m, e_j, q_s), \dots, (q_p, e_n, q_r) \in \delta_{\mathcal{A}_k}$  such that  $w^{Obs} = w'_1 w'_2 \dots w'_{|w^{Obs}|}$ ;

The probability of generating such a path is:

$$P(\theta|\mathcal{A}_k) = I(q_l) \times \prod_{k=1}^{|w^{Obs}|} P(s_{j-1}, w'_j, s_j), \quad (4.8)$$

where  $s_i$  is the  $j^{th}$  state considered in  $\theta(s_0 = q_l, s_1 = q_m, \dots)$ .

Some general considerations are needed to develop the likelihood definition:

A valid path in a PFA  $\mathcal{A}_k$  is a path for some  $w^{Obs} \in \Sigma_{\mathcal{A}_k}^*$  with probability greater than zero. The set of valid paths in  $\mathcal{A}_k$  is denoted  $\Theta_{\mathcal{A}_k}$ .

In general, a given string  $w^{Obs}$  can be generated by  $\mathcal{A}_k$  through multiple valid paths. Let  $\Theta_{\mathcal{A}_k}(w^{Obs})$  denote the set of all the valid paths for  $w^{Obs}$  in  $\mathcal{A}_k$ .

The probability of generating  $w^{Obs}$  with  $\mathcal{A}_k$ , also call likelihood of  $w^{Obs}$  in  $\mathcal{A}_k$  is:

**Definition 4.3** (Likelihood).

$$P(w^{Obs}|\mathcal{A}_k) = \sum_{\theta \in \Theta_{\mathcal{A}_k}(w^{Obs})} P(\theta|\mathcal{A}_k) \quad (4.9)$$

The higher the likelihood, the more probable to perform the activity  $\mathcal{A}_k$ . Let remark here that the value of the likelihood decreases when the length of the considered sequence  $w^{Obs}$  increases. Indeed, let  $t \in \Sigma_{\mathcal{A}_k}^*$  be a sequence composed of events included in the alphabet  $\Sigma_{\mathcal{A}_k}$  and  $e_i \in \Sigma_{\mathcal{A}_k}$  be an event of the same alphabet. Thus:

$$P(te_i|\mathcal{A}_k) \leq P(t|\mathcal{A}_k), \quad (4.10)$$

since, by definition, each valid path  $\theta(te_i)$  for  $te_i$  could be described as a valid path  $\theta(t)$  for  $t$  followed by a transition generating  $e_i$ :

$$\forall \theta(te_i) = (q_l, t'_1, q_m, \dots, q_s, t'_{|t|}, q_p, e_i, q_r); \exists \theta(t) = (q_l, t'_1, q_m, \dots, q_s, t'_{|t|}, q_p), \quad (4.11)$$

with  $\theta(te_i)$  and  $\theta(t)$  two valid path for  $te_i$  and  $t$  in  $\mathcal{A}_k$ .

Then,

$$P(\theta(te_i)|\mathcal{A}_k) = P(\theta(t)|\mathcal{A}_k) \times P(q_p, e_i, q_r) \leq P(\theta(t)|\mathcal{A}_k) \quad (4.12)$$

Finally, the equation (3.2) bounds the summation of the additional transitions  $P(q_p, e_i, q_r)$  to one. Equation (4.10) is thus demonstrated.

The presented *likelihood* computes the distance between a sequence and a PFA and can be extended to compute the distance between a language (i.e. a set of sequences) and a PFA. The major extension of the *likelihood* to languages is the *perplexity*.

### 4.3.2 The perplexity

According to [Jelinek \(1997\)](#), a measure of the complexity of a language considering a model is the mathematical quantity known as *language perplexity*. We can compute the language perplexity, by first defining the entropy  $\widehat{X}(\mathcal{L}_{p_k}^{Obs}|\mathcal{A}_k)$  ([Cover and Thomas, 1991](#)) of the language  $\mathcal{L}_{p_k}^{Obs}$  with respect to  $\mathcal{A}_k$  as given in [Vidal et al. \(2005a\)](#):

**Definition 4.4** (Entropy).

$$\widehat{X}(\mathcal{L}_{p_k}^{Obs}|\mathcal{A}_k) = -\frac{1}{\text{norm}(\mathcal{L}_{p_k}^{Obs})} \times \sum_{w^{Obs} \in \mathcal{L}_{p_k}^{Obs}} \log(P(w^{Obs}|\mathcal{A}_k)), \quad (4.13)$$

with two possible definitions of  $\text{norm}(\mathcal{L}_{p_k}^{Obs})$ :

**Definition 4.5** (Language norm). *The norm  $\text{norm}(\mathcal{L}_{p_k}^{Obs})$  of a language can be defined by considering:*

- *the number of sequences in the language:  $\text{norm}(\mathcal{L}_{p_k}^{Obs}) = \text{card}(\mathcal{L}_{p_k}^{Obs})$ . In this case, the normed is called by string;*
- *the length of the sequences in the language:  $\text{norm}(\mathcal{L}_{p_k}^{Obs}) = \sum_{w^{Obs} \in \mathcal{L}_{p_k}^{Obs}} |w^{Obs}|$ . In this case, the normed is called by symbol;*

The perplexity, denoted  $PP(\mathcal{L}_{p_k}^{Obs}|\mathcal{A}_k)$ , of the language  $\mathcal{L}_{p_k}^{Obs}$  for the PFA  $\mathcal{A}_k$  is then defined as:

**Definition 4.6** (Perplexity).

$$PP(\mathcal{L}_{p_k}^{Obs}|\mathcal{A}_k) = 2^{\widehat{X}(\mathcal{L}_{p_k}^{Obs}|\mathcal{A}_k)}, \quad (4.14)$$

which can also be written as:

$$PP(\mathcal{L}_{p_k}^{Obs}|\mathcal{A}_k) = \left[ \prod_{w^{Obs} \in \mathcal{L}_{p_k}^{Obs}} P(w^{Obs}|\mathcal{A}_k) \right]^{-\frac{1}{\text{norm}(\mathcal{L}_{p_k}^{Obs})}} \quad (4.15)$$

with  $\text{norm}(\mathcal{L}_{p_k}^{Obs})$  as defined in definition 4.5.

The *perplexity* can be seen as the inverse of the *likelihood* geometrical means of sequences in the concerned language. Due to this link with the *likelihood*, the *perplexity* is also strongly linked to the length of the sequences contained in the language  $\mathcal{L}_{p_k}^{Obs}$ .

The *perplexity* has a value between one and the infinity. The lower the *perplexity*, the more probable to perform the activity  $\mathcal{A}_k$ . In order to avoid treating with the infinity, the inverse of the *perplexity* can be preferred.

The *likelihood* and the *perplexity* can be used to detect, among several models, which one has the most likely generated a given sequence. If used to this aim, several hypotheses have to be verified. These hypothesis and their incompatibility to our AR problem are presented in the next subsection 4.3.3.

### 4.3.3 Issues due to distance hypotheses

As demonstrated before, the likelihood and thus the perplexity values are strongly linked to the length of the sequences given as input. Indeed, as shown in equation (4.10), the longest is the sequence, the lower is the likelihood (resp. the higher is the perplexity). Therefore, when used to choose which activity is performed, these distances have to be applied to the same sequence. Furthermore, by applying the likelihood definition, when a noisy event occurs, the likelihood (resp. the perplexity) decreases to zero (resp. diverge to the infinity). Unfortunately in our case, as explained in section 4.2.2, the noise due to several causes (human behaviour, other activities) is unavoidable without to envisage a projection. We are thus in front of a paradoxical issue if we want to use the presented distances. Indeed, according to our choice:

- We do not project the observed events and the obtained distances are useless since always diverging,
- We project the observed events and we compare two incomparable values since obtained by using sequences of different length.

The trouble with the second item is that, due to the property written in equation (4.10), an activity having a low projected sequence may have a better result than another activity having a long and representative projected sequence.

To summarise, the use of the likelihood (resp. perplexity) to compare the probability, for several PFAs, to generate a sequence (resp. a language) is applicable only if:

- the sequences considered for each PFA do not include noise,
- the sequences considered for each PFA have exactly the same length.

Those two points can be systematically true only if, the sequence to compare is the same for all the PFAs and does not include noisy events. Thus, the alphabet of the studied sequence  $w^{Obs}$  have to be included in all PFA alphabets ( $\Sigma_w^{Obs} \in \Sigma_{\mathcal{A}_k} \forall k$ ).

In this presented thesis, it is not the case since activities can be realised in different areas of the smart home and using different tools. Therefore the projections of the sequences, as presented 4.2.2, are unavoidable for us.

As we cannot use the *likelihood* and the *perplexity* like presented due to their strong dependency with the length of the sequence to compare, we need to create new distances based on them but less dependent on the length of the sequences. Hereafter, the *likelihood* and *perplexity* presented in this section 4.3 are called *classical likelihood* and *classical perplexity*.

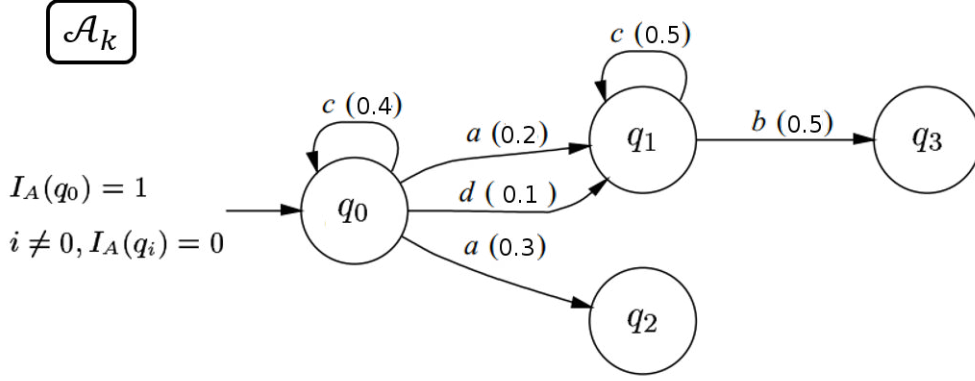
## 4.4 Normalised likelihood and extension

Likelihood based distances are too much dependent on the length of the input data to be used with our work. Indeed, our activity models  $\mathcal{A}_k$  have different alphabets  $\Sigma_{\mathcal{A}_k}$  that can be totally disjoint. This characteristic of our models leads us to project the observed sequences before to perform the language generation and the distance computation. By performing this projection and for the reasons explained before, the existing methods became inapplicable. The objective of this part is to present and develop a new distance between a sequence and a PFA called *normalised likelihood* and its extension to the languages called *normalised perplexity*.

### 4.4.1 The normalised likelihood

In order to be able to choose a model by using likelihood based methods, a normalisation of the classical likelihood is proposed. The objective here is to have a probability really contained between zero and one whatever is the length of the windows. It can be obtained by dividing the classical likelihood by its maximal value obtainable with the considered automaton and with a sequence having the same length as the considered sequence. To illustrate the objective, let's consider the simple PFA  $\mathcal{A}_k$  in figure 4.4.1.

In this figure, for a sequence of length one, the likelihood values are  $P(a|\mathcal{A}_k) = 0.2 + 0.3 = 0.5$ ,  $P(b|\mathcal{A}_k) = 0$ ,  $P(c|\mathcal{A}_k) = 0.4$  and  $P(d|\mathcal{A}_k) = 0.1$ . Thus, the likelihood for a sequence of length one in the example automaton  $\mathcal{A}_k$  is bounded by 0.5: the maximum possible value. In the same way, the likelihood for a sequence of length two in the example automaton  $\mathcal{A}_k$  is bounded by 0.2 (corresponding to  $P(ca|\mathcal{A}_k)$ ). The goal of the normalised likelihood is to have a maximum bound to 1 whatever the length of


 Figure 4.4.1: Example on a simple PFA  $\mathcal{A}_k$ 

the sequence. It works if we divide each likelihood of sequence of length one by 0.5 all sequences of length two by 0.2 and, more generally, all sequences  $w'$  of length  $|w^{Obs}|$  by the maximum value taken by  $P(w'|\mathcal{A}_k)$ .

More formally, the normalised likelihood is defined in Viard et al. (2017) as follows:

**Definition 4.7** (Normalised likelihood). *Let us consider the PFA  $\mathcal{A}_k$  and a given sequence  $w^{Obs} \in \Sigma_{\mathcal{A}_k}^*$ . The normalised likelihood of sequence  $w^{Obs}$  in  $\mathcal{A}_k$ , is defined as:*

$$\|P(w^{Obs}|\mathcal{A}_k)\| = \frac{P(w^{Obs}|\mathcal{A}_k)}{\max_{v \in \Sigma_{\mathcal{A}_k}^{|w^{Obs}|}} [P(v|\mathcal{A}_k)]}. \quad (4.16)$$

with  $P(w^{Obs}|\mathcal{A}_k)$  the *classical likelihood*.

As the *classical likelihood* can be extended to define the *entropy* then the *perplexity*, the *normalised likelihood* can be extended to define the *normalised entropy* then and the *normalised perplexity*. The next subsection will define it.

#### 4.4.2 The normalised perplexity

As the use of the *normalised likelihood* solves the dependency issue of the *classical likelihood* concerning the length of the input sequences, we can define the *normalised entropy* as follows:

**Definition 4.8** (Normalised entropy).

$$\|\widehat{X}(\mathcal{L}_{p_k}^{Obs}|\mathcal{A}_k)\| = -\frac{1}{\text{norm}(\mathcal{L}_{p_k}^{Obs})} \times \sum_{w^{Obs} \in \mathcal{L}_{p_k}^{Obs}} \log(\|P(w^{Obs}|\mathcal{A}_k)\|), \quad (4.17)$$

with  $\text{norm}(\mathcal{L}_{p_k}^{Obs})$  as defined in definition 4.5.

The *normalised perplexity*, denoted  $\|PP(\mathcal{L}_{p_k}^{Obs}|\mathcal{A}_k)\|$ , of the language  $\mathcal{L}_{p_k}^{Obs}$  for the PFA  $\mathcal{A}_k$  is then defined as:

**Definition 4.9** (Normalised perplexity).

$$\|PP(\mathcal{L}_{p_k}^{Obs}|\mathcal{A}_k)\| = 2^{\|\widehat{X}(\mathcal{L}_{p_k}^{Obs}|\mathcal{A}_k)\|}, \quad (4.18)$$

which can also be written as:

$$\|PP(\mathcal{L}_{p_k}^{Obs}|\mathcal{A}_k)\| = \left[ \prod_{w^{Obs} \in \mathcal{L}_{p_k}^{Obs}} \|P(w^{Obs}|\mathcal{A}_k)\| \right]^{-\frac{1}{norm(\mathcal{L}_{p_k}^{Obs})}} \quad (4.19)$$

with  $norm(\mathcal{L}_{p_k}^{Obs})$  as defined in definition 4.5.

As for the *classical entropy* and *normalised likelihood*, the *normalised perplexity* can be seen as the inverse of the *normalised likelihood* geometrical mean of sequences in the concerned language.

As the normalised perplexity represents the distance between a language and its probability to be generated by a model, its value can tend towards infinity and have to be minimised. While the inverse of the normalised perplexity represents the probability, for the considered model, to generate the considered sequence. It is bounded by 1 and has to be maximised. In order to simplify the computation, the inverse of the normalised perplexity  $\frac{1}{\|PP(\mathcal{L}_{p_k}^{Obs}|\mathcal{A}_k)\|}$  is preferred.

### 4.4.3 Computational Issues

The computation of the *normalised likelihood* for an observed sequence  $w^{Obs}$  of length  $|w^{Obs}|$  includes the computation of the maximum value of the *classical likelihood* applied to  $card(\Sigma_{\mathcal{A}_k})^{|w^{Obs}|}$  possible sequences of length  $|w^{Obs}|$ . Its complexity can be written:

$$C_{||L||} = C_L + C_M, \quad (4.20)$$

with  $C_{||L||}$  the complexity of the *normalised likelihood*,  $C_L$  the complexity of the *classical likelihood* and  $C_M$  the computation of the maximum value of the classical likelihood for a length fixed. The normalised likelihood can thus be decomposed in two parts corresponding to two different steps of the computation:

1.  $P(w^{Obs}|\mathcal{A}_k)$  can be computed on-line when the distance estimation is done;
2.  $\max_{v \in \Sigma_{\mathcal{A}_k}^{|w^{Obs}|}} [P(v|\mathcal{A}_k)]$  can be performed off-line for each PFA  $\mathcal{A}_k$  and for all possible values of  $|w^{Obs}|$ .

Even if the forward algorithm (Jurafsky and Martin, 2014) optimises the computation of the classical likelihood to reach the complexity polynomial with  $|w^{Obs}|$

$$C_L = O(card(Q_{\mathcal{A}_k})^2 \times |w^{Obs}|), \quad (4.21)$$

the computation of  $\max_{v \in \Sigma_{\mathcal{A}_k}^{|w^{Obs}|}} [P(v|\mathcal{A}_k)]$  is exponential with  $|w^{Obs}|$  and polynomial with the event set cardinality:

$$C_M = O\left(card(\Sigma_{\mathcal{A}_k})^{|w^{Obs}|} \times C_L\right) = O\left(card(\Sigma_{\mathcal{A}_k})^{|w^{Obs}|} \times card(Q_{\mathcal{A}_k})^2 \times |w^{Obs}|\right). \quad (4.22)$$

Therefore, even if the maximum likelihood is computed off-line, the computational effort may be too high and the complexity of this has to be reduced. To this aim,



we developed two complementary methods of complexity reduction that we will now present:

- the model reduction;
- the dynamical computation.

**Complexity reduction by model reduction:**

Let  $\mathcal{A}_k = \langle Q_{\mathcal{A}_k}, \Sigma_{\mathcal{A}_k}, \delta_{\mathcal{A}_k}, I_{\mathcal{A}_k}, F_{\mathcal{A}_k}, P_{\mathcal{A}_k} \rangle$  be a PFA, we denote by:

$$\mathcal{A}_k^r = \langle Q_{\mathcal{A}_k}, \Sigma_{\mathcal{A}_k}^r, \delta_{\mathcal{A}_k}^r, I_{\mathcal{A}_k}, F_{\mathcal{A}_k}, P_{\mathcal{A}_k}^r \rangle \quad (4.23)$$

the *reduced PFA* associated with  $\mathcal{A}_k$  where  $\Sigma_{\mathcal{A}_k}^r$ ,  $\delta_{\mathcal{A}_k}^r$  and  $P_{\mathcal{A}_k}^r$  are obtained by the *reduction procedure*  $\mathcal{A}_k \rightarrow \mathcal{A}_k^r$ . In order to specify the *reduction procedure*  $\mathcal{A}_k \rightarrow \mathcal{A}_k^r$ , the following definitions are necessary.

First, we denote by  $Geq_{q_g, q_h}(e_i)$  (resp.  $Equ_{q_g, q_h}(e_i)$ ) the set of events  $e_j \in \Sigma_{\mathcal{A}_k}$  having a probability  $P(q_g, e_j, q_h)$  to occur from state  $q_g$  to state  $q_h$  greater than or equal to (resp. equal to)  $e_i \in \Sigma_{\mathcal{A}_k}$  having a probability  $P(q_g, e_i, q_h)$  to occur.

More formally, it holds:

$$Geq_{q_g, q_h}(e_i) = \{e_j | e_j \in \Sigma_{\mathcal{A}_k} \text{ and } P(q_g, e_j, q_h) \geq P(q_g, e_i, q_h)\}, \quad (4.24)$$

and

$$Equ_{q_g, q_h}(e_i) = \{e_j | e_j \in \Sigma_{\mathcal{A}_k} \text{ and } P(q_g, e_j, q_h) = P(q_g, e_i, q_h)\}. \quad (4.25)$$

In the following, the *reduction procedure*  $\mathcal{A}_k \rightarrow \mathcal{A}_k^r$  is presented.

*Step 1: Selection of candidate events*

$$\Sigma_{\mathcal{A}_k}^r = \left\{ e_j \mid e_j \in \Sigma_{\mathcal{A}_k} \text{ and } \bigcap_{q_g, q_h \in Q_{\mathcal{A}_k}} Geq_{q_g, q_h}(e_j) = \bigcap_{q_g, q_h \in Q_{\mathcal{A}_k}} Equ_{q_g, q_h}(e_j) \right\}. \quad (4.26)$$

*Step 2: Deletion of equivalent events*

For all the event sets  $\bigcap_{q_g, q_h \in Q_{\mathcal{A}_k}} Equ_{q_g, q_h}(e_j)$ , only one event is kept: a new  $\Sigma_{\mathcal{A}_k}^r$  is thus obtained.

*Step 3: Conservation of transitions linked to the kept events*

Only transitions implying kept events are conserved, all the others are deleted. Probability of those transitions are not changed. Thus, we have:

$$\delta_{\mathcal{A}_k}^r = \left\{ (q_g, e_j, q_h) \mid (q_g, e_j, q_h) \in \delta_{\mathcal{A}_k} \text{ and } e_j \in \Sigma_{\mathcal{A}_k}^r \right\}, \quad (4.27)$$

and

$$P_{\mathcal{A}_k}^r = \left\{ P(q_g, e_j, q_h) \mid (q_g, e_j, q_h) \in \delta_{\mathcal{A}_k}^r \text{ and } P(q_g, e_j, q_h) \in P_{\mathcal{A}_k} \right\}. \quad (4.28)$$

In Step 1, events having the same intersection sets of  $Geq$  and  $Equ$  are selected, i.e., events having no event with probability to occur greater than one or more transitions and equals for others. In Step 2, the alternatives for sequences having the same probability are deleted.

The problem reduction leads to a new model with a lower number of events than the original one. Thus, the number of combinations to compare in the determination of the

normalised perplexity is reduced. The following proposition proves that the likelihood maximum value is conserved after the reduction.

The figure 4.4.2 shows the *reduced PFA* associated with the PFA  $\mathcal{A}_1$  shown figure 3.3.4. It is obvious that there is less paths possible in this reduced model than in the global model.

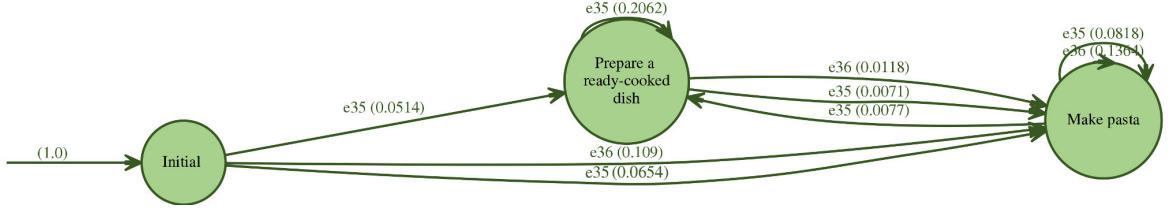


Figure 4.4.2: *Reduced PFA* associated with the PFA  $\mathcal{A}_1$  (figure 3.3.4).

### Proposition 4.1.

Let  $\mathcal{A}_k = \langle Q_{\mathcal{A}_k}, \Sigma_{\mathcal{A}_k}, \delta_{\mathcal{A}_k}, I_{\mathcal{A}_k}, F_{\mathcal{A}_k}, P_{\mathcal{A}_k} \rangle$  be a PFA and  $\mathcal{A}_k^r = \langle Q_{\mathcal{A}_k}, \Sigma_{\mathcal{A}_k}^r, \delta_{\mathcal{A}_k}^r, I_{\mathcal{A}_k}, F_{\mathcal{A}_k}, P_{\mathcal{A}_k}^r \rangle$  be the reduced PFA obtained by the reduction procedure  $\mathcal{A}_k \rightarrow \mathcal{A}_k^r$ . Then  $\forall w^{Obs} \in \Sigma_{\mathcal{A}_k}^*$  of length  $|w^{Obs}|$  it holds:

$$\max_{u \in \Sigma_{\mathcal{A}_k^r}^{|w^{Obs}|}} [P(u|\mathcal{A}_k)] = \max_{v \in \Sigma_{\mathcal{A}_k}^{|w^{Obs}|}} [P(v|\mathcal{A}_k)]. \quad (4.29)$$

The proof is in Appendix A.

By applying the reduction procedure and by Proposition 4.1, the computational complexity of the maximum classical likelihood is reduced as follows:

$$C_M = O \left( \text{card}(\Sigma_{\mathcal{A}_k}^r)^{|w^{Obs}|} \times \text{card}(Q_{\mathcal{A}_k})^2 \times |w^{Obs}| \right). \quad (4.30)$$

Now, the following proposition shows how the complexity (4.30) is reduced if the reduction procedure is used.

### Proposition 4.2.

Let  $\mathcal{A}_k = \langle Q_{\mathcal{A}_k}, \Sigma_{\mathcal{A}_k}, \delta_{\mathcal{A}_k}, I_{\mathcal{A}_k}, F_{\mathcal{A}_k}, P_{\mathcal{A}_k} \rangle$  be a PFA and  $\mathcal{A}_k^r = \langle Q_{\mathcal{A}_k}, \Sigma_{\mathcal{A}_k}^r, \delta_{\mathcal{A}_k}^r, I_{\mathcal{A}_k}, F_{\mathcal{A}_k}, P_{\mathcal{A}_k}^r \rangle$  be the reduced PFA obtained by the reduction procedure  $\mathcal{A}_k \rightarrow \mathcal{A}_k^r$ . Then the computational complexity of the maximum classical likelihood is the following:

$$C_M = O \left( 2^{[\text{card}(Q_{\mathcal{A}_k})-1]^{|w^{Obs}|}} \times \text{card}(Q_{\mathcal{A}_k})^2 \times |w^{Obs}| \right). \quad (4.31)$$

The proof is in Appendix A.

It is important to highlight that in the majority of practical cases, a unique event appears to be maximum for several group of states, decreasing naturally the number of kept events.

The algorithm 5 shows in pseudo-code a function  $GeqEqu$  generating  $Geq_{q_g, q_h}$  (*event*) and  $Equ_{q_g, q_h}$  (*event*) and the model reduction. The complexity  $C_{GeqEqu}$  of the function  $GeqEqu$  is:

$$C_{GeqEqu} = O(\text{card}(\Sigma_{\mathcal{A}_k})). \quad (4.32)$$

and the global model reduction time complexity  $C_{model\_reduction}$  is polynomial:

$$C_{model\_reduction} = O\left(\text{card}(\Sigma_{\mathcal{A}_k})^2 \times \text{card}(Q_{\mathcal{A}_k})^2\right). \quad (4.33)$$

This reduction time complexity is negligible in comparison with the computation time earned by using the reduced PFA  $\mathcal{A}_k^r$  instead of the complete one  $\mathcal{A}_k$ .

---

**Algorithm 5** Model reduction.

---

**Require:** A PFA  $\mathcal{A}_k = \langle Q_{\mathcal{A}_k}, \Sigma_{\mathcal{A}_k}, \delta_{\mathcal{A}_k}, I_{\mathcal{A}_k}, F_{\mathcal{A}_k}, P_{\mathcal{A}_k} \rangle$ .

**Ensure:** A reduced PFA  $\mathcal{A}_k^r = \langle Q_{\mathcal{A}_k}, \Sigma_{\mathcal{A}_k}^r, \delta_{\mathcal{A}_k}^r, I_{\mathcal{A}_k}, F_{\mathcal{A}_k}, P_{\mathcal{A}_k}^r \rangle$ .

```

1: // GeqEqu function. Inputs: a PFA  $\mathcal{A}_k$ , two states  $q_g, q_h \in Q_{\mathcal{A}_k}$  and an event
   event  $\in \Sigma_{\mathcal{A}_k}$ ; Outputs: two lists listGeq and listEqu giving  $\text{Geq}_{q_g, q_h}(\text{event})$  and
    $\text{Equ}_{q_g, q_h}(\text{event})$  respectively.
2: function GeqEqu ( $\mathcal{A}_k, q_g, q_h, \text{event}$ )
3:   // Function initialisation:
4:   listGeq :=  $\emptyset$ 
5:   listEqu :=  $\emptyset$ 
6:   if ( $q_g, \text{event}, q_h$ )  $\in \delta_{\mathcal{A}_k}$  then
7:     proba_ref :=  $P(q_g, \text{event}, q_h)$ 
8:   else
9:     proba_ref := 0
10:  end if
11:  // listGeq Creation:
12:  for all other_event  $\in \Sigma_{\mathcal{A}_k}$  do
13:    if ( $q_g, \text{other\_event}, q_h$ )  $\in \delta_{\mathcal{A}_k}$  then
14:      proba_other :=  $P(q_g, \text{other\_event}, q_h)$ 
15:    else
16:      proba_other := 0
17:    end if
18:    if proba_other  $\geq$  proba_ref then
19:      append other_event to listGeq
20:    end if
21:    if proba_other = proba_ref then
22:      append other_event to listEqu
23:    end if
24:  end for
25:  return listEqu, listGeq
26: end function
27: // Model reduction:
28: // Initialisation:
29:  $\Sigma^r := \emptyset$ 
30: GeqTot :=  $\emptyset$  // GeqTot is a dictionary with keys  $e_j$  and values  $\bigcap_{q_g, q_h \in Q_{\mathcal{A}_k}} \text{Geq}_{q_g, q_h}(e_j)$ .
31: EquTot :=  $\emptyset$  // EquTot is a dictionary with keys  $e_j$  and values  $\bigcap_{q_g, q_h \in Q_{\mathcal{A}_k}} \text{Equ}_{q_g, q_h}(e_j)$ 

```

---

---

```

32: // Computation of GeqTot and EquTot:
33: for all  $e_j \in \Sigma_{\mathcal{A}_k}$  do
34:   // GeqTot[ $e_j$ ] and EquTot[ $e_j$ ] are initialised to  $\Sigma_{\mathcal{A}_k}$  then reduced by intersections.
35:   GeqTot[ $e_j$ ] :=  $\Sigma_{\mathcal{A}_k}$ 
36:   EquTot[ $e_j$ ] :=  $\Sigma_{\mathcal{A}_k}$ 
37:   for all begin_action  $\in Q_{\mathcal{A}_k}$  do
38:     for all end_action  $\in Q_{\mathcal{A}_k} - \text{Initial}$  do
39:       listEqu, listGeq = GeqEqu( $\mathcal{A}_k, \text{begin\_action}, \text{end\_action}, e_j$ )
40:       GeqTot[ $e_j$ ] := GeqTot[ $e_j$ ]  $\cap$  listGeq
41:       EquTot[ $e_j$ ] := EquTot[ $e_j$ ]  $\cap$  listEqu
42:     end for
43:   end for
44: end for
45: // Step 1: selection of candidate events:
46: candidates =  $\emptyset$ 
47: for all  $e_j \in \Sigma_{\mathcal{A}_k}$  do
48:   if then GeqTot[ $e_j$ ] = EquTot[ $e_j$ ]
49:     append  $e_j$  to candidates
50:   end if
51: end for
52: // Step 2: deletion of equivalent events:
53: rejected =  $\emptyset$ 
54: for all  $e_j \in \text{candidates}$  do
55:   if  $e_j \notin \text{rejected}$  then:
56:     append  $e_j$  to  $\Sigma_{\mathcal{A}_k}^r$ 
57:     for all  $e_i \in \text{EquTot}[e_j]$  do
58:       if  $e_i \neq e_j$  then
59:         append  $e_i$  to rejected
60:       end if
61:     end for
62:   end if
63: end for
64: // Step 3: conservation of transitions linked to the kept events:
65:  $\delta_{\mathcal{A}_k}^r = \emptyset$ 
66:  $P_{\mathcal{A}_k}^r = \emptyset$ 
67: for all  $\text{do}(q_h, e_i, q_g) \in \delta_{\mathcal{A}_k}$ 
68:   if  $e_i \in \Sigma_{\mathcal{A}_k}^r$  then
69:     add  $(q_h, e_i, q_g)$  to  $\delta_{\mathcal{A}_k}^r$ 
70:     add  $P(q_h, e_i, q_g)$  from  $P_{\mathcal{A}_k}$  to  $P_{\mathcal{A}_k}^r$ 
71:   end if
72: end for

```

---

### Complexity reduction by computational simplification:

In addition to the model reduction, a computational simplification can be employed. Indeed, to compute the  $\max_{v \in \Sigma_{\mathcal{A}_k}^{|w, \text{Obs}|}} [P(v|\mathcal{A}_k)]$ , a natural idea is to independently compute the value of  $P(v|\mathcal{A}_k)$  for all possible  $v$  by applying the forward algorithm (Jurafsky and Martin, 2014) that we will describe now as in Vidal et al. (2005a).

The probabilities assigned to the paths in  $\Theta_{\mathcal{A}_k}$  (section 4.3.1) are used to compute the *classical likelihood* efficiently  $P(w^{Obs}|\mathcal{A}_k)$ . The idea is similar to the one proposed for HMMs (Ney, 1997) by defining  $\alpha_{w_1\dots w_l}(i)\forall i \in 0 \leq i \leq \text{card}(Q_{\mathcal{A}_k}) - 1$  as the probability of generating the sequence  $w_1\dots w_l$  of length  $l$ , prefix of  $w^{Obs}$  and reaching state  $q_i$ :

$$\alpha_{w_1\dots w_l}(i) = \sum_{\substack{(s_0, s_1, \dots, s_l) \in \Theta_{\mathcal{A}_k}(w_1\dots w_l) \\ s_l = q_i}} I(s_0) \times \prod_{j=1}^l P(s_{j-1}, w_j, s_j), \quad (4.34)$$

Equation (4.34) can be calculated with the following algorithm:

*Forward algorithm:*

$$\begin{aligned} \alpha_\epsilon(i) &= I(q_i), \\ \alpha_{w_1\dots w_j}(i) &= \sum_{q_g \in Q_{\mathcal{A}_k}} \alpha_{w_1\dots w_{j-1}}(g) \times P(q_g, w_j, q_i), 1 \leq j \leq |w^{Obs}| \end{aligned} \quad (4.35)$$

For a sequence  $w^{Obs} \in \Sigma_{\mathcal{A}_k}^*$ , the following proposition is straightforward:

**Proposition 4.3.**

$$P(w^{Obs}|\mathcal{A}_k) = \sum_{q_i \in Q_{\mathcal{A}_k}} \alpha_w^{Obs}(i). \quad (4.36)$$

Therefore, the computation of  $P(w^{Obs}|\mathcal{A}_k)$  can be performed with a time complexity of  $O(|w^{Obs}| \times \text{card}(Q_{\mathcal{A}_k})^2)$  where  $|w^{Obs}|$  is the length of  $w^{Obs}$  and  $\text{card}(Q_{\mathcal{A}_k})$  the number of states in  $\mathcal{A}_k$ . Algorithm 6 shows this *classical likelihood* computation.

---

**Algorithm 6** Forward algorithm.

---

**Require:**

- A PFA  $\mathcal{A}_k = \langle Q_{\mathcal{A}_k}, \Sigma_{\mathcal{A}_k}, \delta_{\mathcal{A}_k}, I_{\mathcal{A}_k}, F_{\mathcal{A}_k}, P_{\mathcal{A}_k} \rangle$ ;
- An observed sequence  $w^{Obs}$ .

**Ensure:** The likelihood  $P(w^{Obs}|\mathcal{A}_k)$ .

- 1: // *alpha function*. Inputs: a sequence  $w$ , a PFA  $\mathcal{A}_k$  and a state index  $i$
  - 2: **function** alpha ( $w, \mathcal{A}_k, i$ )
  - 3:   **if**  $|w| > 0$  **then**
  - 4:      $result := \sum_{q_g \in Q_{\mathcal{A}_k}} \text{alpha}(w_1\dots w_{|w|-1}, \mathcal{A}_k, g) \times P(q_g, w_j, q_i)$
  - 5:   **else**
  - 6:      $result := I(q_i)$
  - 7:   **end if**
  - 8:   **return** result
  - 9: **end function**
  - 10:  $P(w^{Obs}|\mathcal{A}_k) = \sum_{q_i \in Q_{\mathcal{A}_k}} \text{alpha}(w^{Obs}, \mathcal{A}_k, i)$ .
- 

Now the forward algorithm known, we can see that, by performing each *classical likelihood* independently to find  $\max_{v \in \Sigma_{\mathcal{A}_k}^{|w^{Obs}|}} [P(v|\mathcal{A}_k)]$ , the intermediate values of  $\alpha_{v_1\dots v_j}(i)$  are computed several times: once for all sequences having  $v_1\dots v_j$  as a prefix, as shown with a simple example in figure 4.4.3.

The idea of the dynamical computation is to avoid multiple computation of the same  $\alpha_{v_1\dots v_j}(i)$  by keeping it in memory as long as needed, see figure 4.4.4.

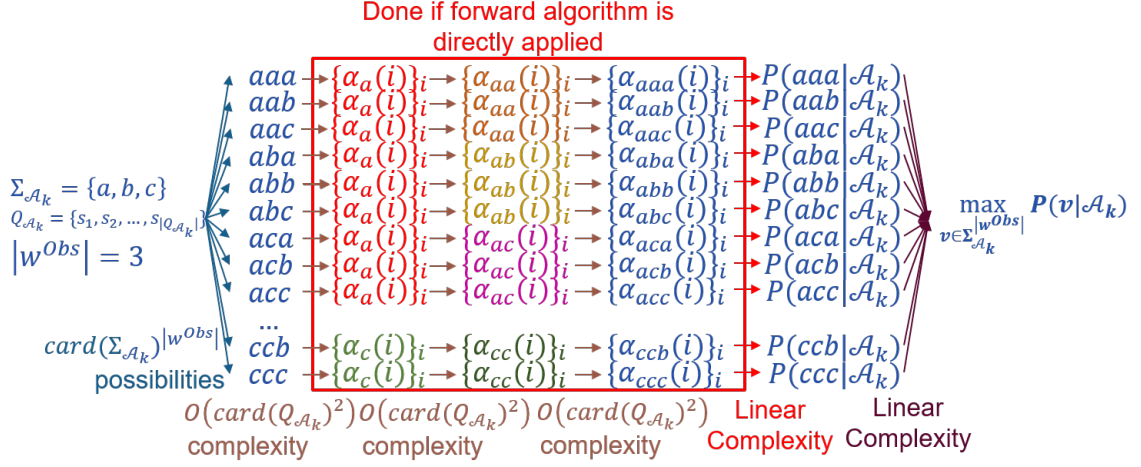


Figure 4.4.3: Computation of the maximum likelihood without dynamical computation.

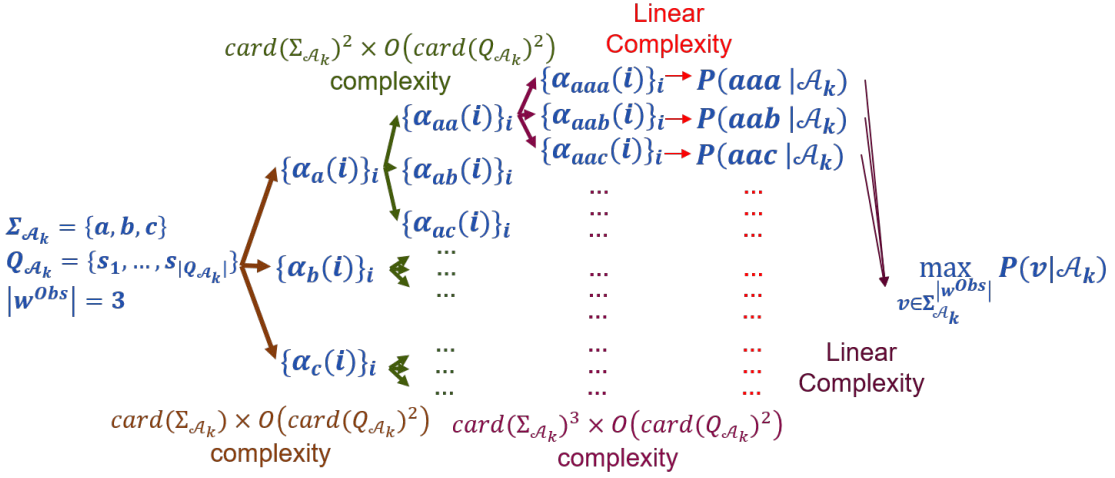


Figure 4.4.4: Computation of the maximum likelihood with dynamical computation

As shown in the figure, the complexity by applying this method is changing from:

$$C_M = O\left(\text{card}(\Sigma_{\mathcal{A}_k})^{|w^{Obs}|} \times \text{card}(Q_{\mathcal{A}_k})^2 \times |w^{Obs}|\right), \quad (4.37)$$

to

$$C_M = O\left(\sum_{l=1}^{|w^{Obs}|} \text{card}(\Sigma_{\mathcal{A}_k})^l \times \text{card}(Q_{\mathcal{A}_k})^2\right) = O\left(\text{card}(\Sigma_{\mathcal{A}_k})^{|w^{Obs}|} \times \text{card}(Q_{\mathcal{A}_k})^2\right), \quad (4.38)$$

Finally, this dynamic reduction shown in algorithm 7, removes the linear component  $|w^{Obs}|$  of the complexity.

---

**Algorithm 7** Dynamical computation of  $\max_{v \in \Sigma_{\mathcal{A}_k}^{|w^{Obs}|}} [P(v|\mathcal{A}_k)]$ .

---

**Require:**

- A PFA  $\mathcal{A}_k = \langle Q_{\mathcal{A}_k}, \Sigma_{\mathcal{A}_k}, \delta_{\mathcal{A}_k}, I_{\mathcal{A}_k}, F_{\mathcal{A}_k}, P_{\mathcal{A}_k} \rangle$ ;
- A sequence length  $|w^{Obs}|$ .

**Ensure:** The maximum likelihood  $\max_{v \in \Sigma_{\mathcal{A}_k}^{|w^{Obs}|}} [P(v|\mathcal{A}_k)]$ .

```

1: // browse function. Inputs: a PFA  $\mathcal{A}_k$ , a sequence length  $length$ , a list  $listAlphas$ 
  of already computed dynamical variables  $\alpha(v, \mathcal{A}_k, i)$  and an integer  $step$  traducing
  the step already reached:
2: function browse ( $\mathcal{A}_k, length, listAlphas, step$ )
3:   likelihoods :=  $\emptyset$  // list in which all maximum likelihoods computed are stocked.
4:   newAlphas :=  $\emptyset$ 
5:   if  $step := 1$  then
6:     for all  $e_j \in \Sigma_{\mathcal{A}_k}$  do
7:       for all  $end\_action \in Q_{\mathcal{A}_k}$  do
8:         append  $I(end\_action)$  to newAlphas
9:       end for
10:      if  $step = length$  then
11:        append  $\Sigma newAlphas$  to likelihoods
12:      else
13:        following_result := browse( $\mathcal{A}_k, |w^{Obs}|, newAlphas, step + 1$ )
14:        append following_result to likelihoods
15:      end if
16:    end for
17:  else
18:    for all  $e_j \in \Sigma_{\mathcal{A}_k}$  do
19:      for all  $end\_action \in Q_{\mathcal{A}_k}$  do
20:        temp_alpha :=  $\sum_{g:=1 \text{ to } card(Q_{\mathcal{A}_k})} listAlphas[g] \times P(q_g, e_j, end\_action)$ 
21:        append temp_alpha to newAlphas
22:      end for
23:      if  $step = length$  then
24:        append  $\sum_{g:=1 \text{ to } card(Q_{\mathcal{A}_k})} new\_Alphas[g]$  to likelihoods
25:      else
26:        following_result := browse( $\mathcal{A}_k, |w^{Obs}|, new\_Alphas, step + 1$ )
27:        append following_result to likelihoods
28:      end if
29:    end for
30:  end if
31:  result :=  $\max(likelihoods)$ 
32:  return result
33: end function
34: // Launch the wanted computation:
35:  $\max_{v \in \Sigma_{\mathcal{A}_k}^{|w^{Obs}|}} [P(v|\mathcal{A}_k)] := browse(\mathcal{A}_k, |w^{Obs}|, \emptyset, 1)$ 

```

---

**To summarise:**

The two presented reduction methods act on two independent points influencing the computation complexity: the model used to compute and the computation algorithm. The table 4.4.1 recalls the complexity of the maximum classical likelihood computation by considering, or not, the different reduction methods. The reduction of the cardinality of the exponential part of the complexity and the removal of the linear component allow a quick computation time compatible with a real-time analysis.

Table 4.4.1: Time complexity of the maximum classical likelihood computation for a given length of sequence equal to  $|w^{Obs}|$  with  $card(Q_{\mathcal{A}_k}) \ll card(\Sigma_{\mathcal{A}_k})$ .

	Without <i>Dynamical calculation</i>	With <i>Dynamical calculation</i>
Without <i>Problem reduction</i>	$O(card(\Sigma_{\mathcal{A}_k})^{ w^{Obs} } \times card(Q_{\mathcal{A}_k})^2 \times  w^{Obs} )$	$O(2^{[card(Q_{\mathcal{A}_k})-1] w^{Obs} } \times card(Q_{\mathcal{A}_k})^2 \times  w^{Obs} )$
With <i>Problem reduction</i>	$O(card(\Sigma_{\mathcal{A}_k})^{ w^{Obs} } \times card(Q_{\mathcal{A}_k})^2)$	$O(2^{[card(Q_{\mathcal{A}_k})-1] w^{Obs} } \times card(Q_{\mathcal{A}_k})^2)$

It is important to underline that in the majority of practical cases, a unique event appears to be maximum for several group of states, decreasing naturally the number of kept events. By experiments, it is the case in the majority of times.

In order to illustrate the presented AR protocol, an application is made in the next section.



## 4.5 Application to the Case Study

In this section, the AR protocol is applied to the study case presented in section 2.2. The recognition during an occurrence of each of the three activities is presented. A way to read the obtained distances is presented and a discussion about result viability and validation is realised in the section 4.6.

### 4.5.1 Observed sequence windowing

In order to illustrate the AR process, the fourth protocol steps are applied to the three activities of our case study:  $\mathcal{A}_1$ : cooking,  $\mathcal{A}_2$ : hot beverage preparation and  $\mathcal{A}_3$ : use bathroom. Activity models used are those presented in figures 3.3.4 to 3.3.6.

As during the learning period, a windowing has to be performed in order to treat a long enough sequence to observe the activity occurrence but not too long to facilitate the computation. Furthermore, as a succession of short activities has to be detected as several activity, and not as a unique activity occurrence, the number of treat events should not be too long. By experimentation, we find that, for our study case log, a length of sequence to observe equal to five is a good trade-off to obtain valid results.

In the section 4.5, the recognition protocol is applied to the same three significant subsequences of life than in section 3.3:  $w_{\mathcal{A}_1}^{Obs}$ ,  $w_{\mathcal{A}_2}^{Obs}$  and  $w_{\mathcal{A}_3}^{Obs}$ . Each chosen subsequence represents a part of an occurrence of activities  $\mathcal{A}_1$ ,  $\mathcal{A}_2$  and  $\mathcal{A}_3$ , respectively.

The chosen example sequences are thus:

- $w_{\mathcal{A}_1}^{Obs} = e_{23} e_{14} e_{13} e_{37} e_{36} e_{23} e_{33} e_{32} e_{35}$
- $w_{\mathcal{A}_2}^{Obs} = e_{36} e_{29} e_{28} e_{23} e_{16} e_{15} e_{18} e_{17} e_{23}$
- $w_{\mathcal{A}_3}^{Obs} = e_{39} e_{38} e_2 e_1 e_4 e_7 e_6 e_5 e_3$

Let recall than the three retained sequences are also representative of three specific situations:

- $w_{\mathcal{A}_1}^{Obs}$  is an activity  $\mathcal{A}_1$  performance of an activity sharing a lot of events with another one ( $w_{\mathcal{A}_2}^{Obs}$ ),
- $w_{\mathcal{A}_2}^{Obs}$  is an activity  $\mathcal{A}_2$  performance of an activity sharing a little part of events with another one ( $w_{\mathcal{A}_1}^{Obs}$ ),
- $w_{\mathcal{A}_3}^{Obs}$  is an activity  $\mathcal{A}_3$  performance of an activity sharing no event with any other ones.

Figure 4.5.1 lists the sequences obtained by keeping the last five observed events. For the sake of simplicity, we consider than the fourth first events already occurred and we show the kept sequence when a new event occurs. In the real case, at the beginning of the recognition, we use all we can. That means that if only three events occur since the beginning of the recognition, the complete sequence is kept. It does not have real importance since the beginning of the recognition represents a very few part of the inhabitant life and the quality of the obtained information at this specific time does not matter since it is a small transition phase for us.

### 4.5.2 Projection of the exploited sequence

The second step of our AR protocol consists in a projection of the fixed length sequences (five here) of events extracted from the sensors information flow. The projection step illustrated here has as impact that all computation performed after it produces results strongly linked to each PFA  $\mathcal{A}_k$ . Thus, result obtained between an input sequence and a PFA depends only on the concerned PFA since computations are independent.

Table 4.5.1: Analysed sequences each time a new event of the sequence  $w_{\mathcal{A}_k}^{Obs}$  occurs.

During the performance of sequence $w_{\mathcal{A}_1}^{Obs} = e_{23} e_{14} e_{13} e_{37} e_{36} e_{23} e_{33} e_{32} e_{35}$				
occurrence of $e_{36}$	occurrence of $e_{23}$	occurrence of $e_{33}$	occurrence of $e_{32}$	occurrence of $e_{35}$
$e_{23} e_{14} e_{13} e_{37} e_{36}$	$e_{14} e_{13} e_{37} e_{36} e_{23}$	$e_{13} e_{37} e_{36} e_{23} e_{33}$	$e_{37} e_{36} e_{23} e_{33} e_{32}$	$e_{36} e_{23} e_{33} e_{32} e_{35}$
During the performance of sequence $w_{\mathcal{A}_2}^{Obs} = e_{36} e_{29} e_{28} e_{23} e_{16} e_{15} e_{18} e_{17} e_{23}$				
occurrence of $e_{16}$	occurrence of $e_{15}$	occurrence of $e_{18}$	occurrence of $e_{17}$	occurrence of $e_{23}$
$e_{36} e_{29} e_{28} e_{23} e_{16}$	$e_{29} e_{28} e_{23} e_{16} e_{15}$	$e_{28} e_{23} e_{16} e_{15} e_{18}$	$e_{23} e_{16} e_{15} e_{18} e_{17}$	$e_{16} e_{15} e_{18} e_{17} e_{23}$
During the performance of sequence $w_{\mathcal{A}_3}^{Obs} = e_{39} e_{38} e_2 e_1 e_4 e_7 e_6 e_5 e_3$				
occurrence of $e_4$	occurrence of $e_7$	occurrence of $e_6$	occurrence of $e_5$	occurrence of $e_3$
$e_{39} e_{38} e_2 e_1 e_4$	$e_{38} e_2 e_1 e_4 e_7$	$e_2 e_1 e_4 e_7 e_6$	$e_1 e_4 e_7 e_6 e_5$	$e_4 e_7 e_6 e_5 e_3$

This computation independence is an advantage in the real life since activity to monitor can evolve with the time according to the pathologies the medical staff want to focus. If the medical staff want to focus on the cooking activity only during a fixed period of the inhabitant life, the monitoring of the other activities can be considered as useless or unwanted. Furthermore, the independence of the distances computation can lead us to indecision since two activities will be considered as possible. In this case, the decision making can be reported to later if the indecision disappears with the time or it can be asked to an expert to distinguish the occurred activity or to improve the instrumentation.

Figure 4.5.2 lists the different projected sequences obtained after each occurrence of a new event with the example sequences  $w_{\mathcal{A}_1}^{Obs}$ ,  $w_{\mathcal{A}_2}^{Obs}$  and  $w_{\mathcal{A}_3}^{Obs}$ . The projection of the analysed sequences obtained above (figure 4.5.1) on the activity alphabet  $\Sigma_{\mathcal{A}_k}$  is denoted  $w_{p_k}^{Obs}$ . The list of events  $\Sigma_{\mathcal{A}_k}$  linked with activity  $\mathcal{A}_k$  can be seen in figures 2.2.10 or 3.3.2 and are the following:

- $\Sigma_{\mathcal{A}_1} = \{e_{13}, e_{14}, e_{21}, e_{22}, e_{24}, e_{25}, e_{26}, e_{27}, e_{28}, e_{29}, e_{30}, e_{31}, e_{32}, e_{33}, e_{34}, e_{35}, e_{36}, e_{37}\}$ ;
- $\Sigma_{\mathcal{A}_2} = \{e_{11}, e_{12}, e_{13}, e_{14}, e_{15}, e_{16}, e_{17}, e_{18}, e_{19}, e_{20}, e_{24}, e_{25}, e_{26}, e_{27}, e_{28}, e_{29}, e_{36}, e_{37}\}$ ;
- $\Sigma_{\mathcal{A}_3} = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}, e_{38}, e_{39}\}$ .

### 4.5.3 Language generation based on the projected sequence

Once the extracted sequence projected, it is necessary to extract a language from the obtained projected sequences  $w_{p_k}^{Obs}$ . Indeed, as explained in section 4.2.2, the language creation is necessary to traduce the continuous property of the human life and to consider all successions of events without any unwanted discrimination.

The retained language generation method is illustrated in figure 4.2.2 and consists in creating a language  $\mathcal{L}_{p_k}^{Obs}$  composed of all substrings of the projected sequence  $w_{p_k}^{Obs}$  having a length greater than or equal to two. Figures 4.5.3 to 4.5.5 shows the languages  $\mathcal{L}_{p_k}^{Obs}$  obtained for the projected sequences  $w_{p_k}^{Obs}$  shown in figure 4.5.2.

Table 4.5.2: Projected sequences  $w_{p_k}^{Obs}$  when new events of the sequences  $w_{A_k}^{Obs}$  occurs.

During the performance of sequence $w_{A_1}^{Obs} = e_{23} e_{14} e_{13} e_{37} e_{36} e_{23} e_{33} e_{32} e_{35}$					
	occurrence of $e_{36}$	occurrence of $e_{23}$	occurrence of $e_{33}$	occurrence of $e_{32}$	occurrence of $e_{35}$
$w_{p_1}^{Obs}$	$e_{14} e_{13} e_{37} e_{36}$	$e_{14} e_{13} e_{37} e_{36}$	$e_{13} e_{37} e_{36} e_{33}$	$e_{37} e_{36} e_{33} e_{32}$	$e_{36} e_{33} e_{32} e_{35}$
$w_{p_2}^{Obs}$	$e_{14} e_{13} e_{37} e_{36}$	$e_{14} e_{13} e_{37} e_{36}$	$e_{13} e_{37} e_{36}$	$e_{37} e_{36}$	$e_{36}$
$w_{p_3}^{Obs}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
During the performance of sequence $w_{A_2}^{Obs} = e_{36} e_{29} e_{28} e_{23} e_{16} e_{15} e_{18} e_{17} e_{23}$					
	occurrence of $e_{16}$	occurrence of $e_{15}$	occurrence of $e_{18}$	occurrence of $e_{17}$	occurrence of $e_{23}$
$w_{p_1}^{Obs}$	$e_{36} e_{29} e_{28}$	$e_{29} e_{28}$	$e_{28}$	$\emptyset$	$\emptyset$
$w_{p_2}^{Obs}$	$e_{36} e_{29} e_{28} e_{16}$	$e_{29} e_{28} e_{16} e_{15}$	$e_{28} e_{16} e_{15} e_{18}$	$e_{16} e_{15} e_{18} e_{17}$	$e_{16} e_{15} e_{18} e_{17}$
$w_{p_3}^{Obs}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
During the performance of sequence $w_{A_3}^{Obs} = e_{39} e_{38} e_2 e_1 e_4 e_7 e_6 e_5 e_3$					
	occurrence of $e_4$	occurrence of $e_7$	occurrence of $e_6$	occurrence of $e_5$	occurrence of $e_3$
$w_{p_1}^{Obs}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
$w_{p_2}^{Obs}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
$w_{p_3}^{Obs}$	$e_{39} e_{38} e_2 e_1 e_4$	$e_{38} e_2 e_1 e_4 e_7$	$e_2 e_1 e_4 e_7 e_6$	$e_1 e_4 e_7 e_6 e_5$	$e_4 e_7 e_6 e_5 e_3$

 Table 4.5.3: Generated languages  $\mathcal{L}_{p_k}^{Obs}$  when new events of the sequence  $w_{A_1}^{Obs}$  occurs.

During the performance of sequence $w_{A_1}^{Obs} = e_{23} e_{14} e_{13} e_{37} e_{36} e_{23} e_{33} e_{32} e_{35}$	
	occurrence of $e_{36}$
$\mathcal{L}_{p_1}^{Obs}$	$\{e_{14}e_{13}, e_{13}e_{37}, e_{37}e_{36}, e_{14}e_{13}e_{37}, e_{13}e_{37}e_{36}, e_{14}e_{13}e_{37}e_{36}\}$
$\mathcal{L}_{p_2}^{Obs}$	$\{e_{14}e_{13}, e_{13}e_{37}, e_{37}e_{36}, e_{14}e_{13}e_{37}, e_{13}e_{37}e_{36}, e_{14}e_{13}e_{37}e_{36}\}$
$\mathcal{L}_{p_3}^{Obs}$	$\emptyset$
	occurrence of $e_{23}$
$\mathcal{L}_{p_1}^{Obs}$	$\{e_{14}e_{13}, e_{13}e_{37}, e_{37}e_{36}, e_{14}e_{13}e_{37}, e_{13}e_{37}e_{36}, e_{14}e_{13}e_{37}e_{36}\}$
$\mathcal{L}_{p_2}^{Obs}$	$\{e_{14}e_{13}, e_{13}e_{37}, e_{37}e_{36}, e_{14}e_{13}e_{37}, e_{13}e_{37}e_{36}, e_{14}e_{13}e_{37}e_{36}\}$
$\mathcal{L}_{p_3}^{Obs}$	$\emptyset$
	occurrence of $e_{33}$
$\mathcal{L}_{p_1}^{Obs}$	$\{e_{13}e_{37}, e_{37}e_{36}, e_{36}e_{33}, e_{13}e_{37}e_{36}, e_{37}e_{36}e_{33}, e_{13}e_{37}e_{36}e_{33}\}$
$\mathcal{L}_{p_2}^{Obs}$	$\{e_{13}e_{37}, e_{37}e_{36}, e_{13}e_{37}e_{36}\}$
$\mathcal{L}_{p_3}^{Obs}$	$\emptyset$
	occurrence of $e_{32}$
$\mathcal{L}_{p_1}^{Obs}$	$\{e_{37}e_{36}, e_{36}e_{33}, e_{33}e_{32}, e_{37}e_{36}e_{33}, e_{36}e_{33}e_{32}, e_{37}e_{36}e_{33}e_{32}\}$
$\mathcal{L}_{p_2}^{Obs}$	$\{e_{37}e_{36}\}$
$\mathcal{L}_{p_3}^{Obs}$	$\emptyset$
	occurrence of $e_{35}$
$\mathcal{L}_{p_1}^{Obs}$	$\{e_{36}e_{33}, e_{33}e_{32}, e_{32}e_{35}, e_{36}e_{33}e_{32}, e_{33}e_{32}e_{35}, e_{36}e_{33}e_{32}e_{35}\}$
$\mathcal{L}_{p_2}^{Obs}$	$\emptyset$
$\mathcal{L}_{p_3}^{Obs}$	$\emptyset$

By studying the languages obtained by adding a unique event, we can see that most of the sequences of the alphabet are kept from one step to another one. Indeed, if we

Table 4.5.4: Generated languages  $\mathcal{L}_{p_k}^{Obs}$  when new events of the sequence  $w_{A_2}^{Obs}$  occurs.

		During the performance of sequence $w_{A_2}^{Obs} = e_{36} e_{29} e_{28} e_{23} e_{16} e_{15} e_{18} e_{17} e_{23}$	
		occurrence of $e_{16}$	occurrence of $e_{15}$
$\mathcal{L}_{p_1}^{Obs}$	-----	$\{e_{36}e_{29}, e_{29}e_{28}, e_{36}e_{29}e_{28}\}$	$\{e_{29}e_{28}\}$
$\mathcal{L}_{p_2}^{Obs}$	-----	$\{e_{36}e_{29}, e_{29}e_{28}, e_{28}e_{16}, e_{36}e_{29}e_{28}, e_{29}e_{28}e_{16}, e_{36}e_{29}e_{28}e_{16}\}$	$\{e_{29}e_{28}, e_{28}e_{16}, e_{16}e_{15}, e_{29}e_{28}e_{16}, e_{28}e_{16}e_{15}, e_{29}e_{28}e_{16}e_{15}\}$
$\mathcal{L}_{p_3}^{Obs}$	-----	$\emptyset$	$\emptyset$
		occurrence of $e_{18}$	occurrence of $e_{17}$
$\mathcal{L}_{p_1}^{Obs}$	-----	$\emptyset$	$\emptyset$
$\mathcal{L}_{p_2}^{Obs}$	-----	$\{e_{28}e_{16}, e_{16}e_{15}, e_{15}e_{18}, e_{28}e_{16}e_{15}, e_{16}e_{15}e_{18}, e_{28}e_{16}e_{15}e_{18}\}$	$\{e_{16}e_{15}, e_{15}e_{18}, e_{18}e_{17}, e_{16}e_{15}e_{18}, e_{15}e_{18}e_{17}, e_{16}e_{15}e_{18}e_{17}\}$
$\mathcal{L}_{p_3}^{Obs}$	-----	$\emptyset$	$\emptyset$
		occurrence of $e_{23}$	
$\mathcal{L}_{p_1}^{Obs}$	-----	$\emptyset$	$\emptyset$
$\mathcal{L}_{p_2}^{Obs}$	-----	$\{e_{16}e_{15}, e_{15}e_{18}, e_{18}e_{17}, e_{16}e_{15}e_{18}, e_{15}e_{18}e_{17}, e_{16}e_{15}e_{18}e_{17}\}$	
$\mathcal{L}_{p_3}^{Obs}$	-----	$\emptyset$	

Table 4.5.5: Generated languages  $\mathcal{L}_{p_k}^{Obs}$  when new events of the sequence  $w_{A_3}^{Obs}$  occurs.

		During the performance of sequence $w_{A_3}^{Obs} = e_{39} e_{38} e_2 e_1 e_4 e_7 e_6 e_5 e_3$	
		occurrence of $e_4$	occurrence of $e_7$
$\mathcal{L}_{p_1}^{Obs}$	-----	$\emptyset$	$\emptyset$
$\mathcal{L}_{p_2}^{Obs}$	-----	$\emptyset$	$\emptyset$
$\mathcal{L}_{p_3}^{Obs}$	-----	$\{e_{39}e_{38}, e_{38}e_2, e_2e_1, e_1e_4, e_{39}e_{38}e_2, e_{38}e_2e_1, e_2e_1e_4, e_{39}e_{38}e_2e_1, e_{38}e_2e_1e_4, e_{39}e_{38}e_2e_1e_4\}$	$\{e_{38}e_2, e_2e_1, e_1e_4, e_4e_7, e_{38}e_2e_1, e_2e_1e_4, e_1e_4e_7, e_{38}e_2e_1e_4, e_2e_1e_4e_7, e_{38}e_2e_1e_4e_7\}$
		occurrence of $e_6$	occurrence of $e_5$
$\mathcal{L}_{p_1}^{Obs}$	-----	$\emptyset$	$\emptyset$
$\mathcal{L}_{p_2}^{Obs}$	-----	$\emptyset$	$\emptyset$
$\mathcal{L}_{p_3}^{Obs}$	-----	$\{e_2e_1, e_1e_4, e_4e_7, e_7e_6, e_2e_1e_4, e_1e_4e_7, e_4e_7e_6, e_2e_1e_4e_7, e_1e_4e_7e_6, e_2e_1e_4e_7e_6\}$	$\{e_1e_4, e_4e_7, e_7e_6, e_6e_5, e_1e_4e_7, e_4e_7e_6, e_7e_6e_5, e_1e_4e_7e_6, e_4e_7e_6e_5, e_1e_4e_7e_6e_5\}$
		occurrence of $e_3$	
$\mathcal{L}_{p_1}^{Obs}$	-----	$\emptyset$	$\emptyset$
$\mathcal{L}_{p_2}^{Obs}$	-----	$\emptyset$	$\emptyset$
$\mathcal{L}_{p_3}^{Obs}$	-----	$\{e_4e_7, e_7e_6, e_6e_5, e_5e_3, e_4e_7e_6, e_7e_6e_5, e_6e_5e_3, e_4e_7e_6e_5, e_7e_6e_5e_3, e_4e_7e_6e_5e_3\}$	

compare alphabets in the figure 4.5.5 just above, we can see that six sequences are kept from the language  $\mathcal{L}_{p_3}^{Obs}$  generated after the occurrence of  $e_4$  to the one generated after the occurrence of  $e_7$ , since four sequences changes. It is exactly what we want since the occurrence of a single event does not change totally what the inhabitant do. At each step, we keep the shared substring between the past and the present to traduce the life continuity: a single event does not totally change the manipulated language. This

property can be seen for all languages and for all events succession. In the same way, when the number of events in the projected sequence decrease, the language progressively lose sequences until it is empty.

#### 4.5.4 Probability estimation

Each time a new event is observed and once the first three steps of the protocol performed, we have one language for each activity  $\mathcal{A}_k$  to monitor. A distance can thus be computed between each PFA modelling those activities  $\{\mathcal{A}_k\}$  and the corresponding languages  $\{\mathcal{L}_{p_k}^{Obs}\}$ . In our case, the retained distance is the inverse of the normalised perplexity. Therefore, as a final step of the AR, the inverse of the normalised perplexity  $\frac{1}{\|PP(\mathcal{L}_{p_k}^{Obs}|\mathcal{A}_k)\|}$  for each model  $\mathcal{A}_k$  to generate the language  $\mathcal{L}_{p_k}^{Obs}$  is computed. Since the differences between perplexity per string or per symbol is only proportional, we prefer to use the perplexity per symbol to underline the importance of the events.

For a sake of simplicity, all the details of the distance computation will not be presented here. The intermediate computation results for  $w^{Obs} = e_{23}e_{14} e_{13} e_{37} e_{36}$  extracted from  $w_{\mathcal{A}_1}^{Obs}$  are given in figure 4.5.6. In this figure, we can find, for each language  $\mathcal{L}_{p_1}^{Obs}$ ,  $\mathcal{L}_{p_2}^{Obs}$  and  $\mathcal{L}_{p_3}^{Obs}$ , the sequences included in the language and their likelihood, maximum likelihood, normalised likelihood and inverse of normalised perplexity per symbol knowing the activity model  $\mathcal{A}_1$ ,  $\mathcal{A}_2$  and  $\mathcal{A}_3$ , respectively.

Table 4.5.6: Pertinent intermediate probability for the distance computation of the example observed sequence and final distance.

	sequence $u$	$P(u \mathcal{A}_k)$	$\max_{v \in \Sigma_{\mathcal{A}_k}^{ u }} [P(v \mathcal{A}_k)]$	$\ P(u \mathcal{A}_k)\ $	$\frac{1}{\ PP(\mathcal{L}_{p_k}^{Obs} \mathcal{A}_k)\ }$
$\mathcal{L}_{p_1}^{Obs}$	$e_{14}e_{13}$	0.01288334	0.01628777	0.79098265	0.86705593
	$e_{13}e_{37}$	0.01360917	0.01628777	0.83554505	
	$e_{37}e_{36}$	0.01427535	0.01628777	0.87644586	
	$e_{14}e_{13}e_{37}$	0.00171671	0.00277459	0.61872418	
	$e_{13}e_{37}e_{36}$	0.00181342	0.00277459	0.65358188	
	$e_{14}e_{13}e_{37}e_{36}$	0.00022875	0.00052513	0.43560692	
$\mathcal{L}_{p_2}^{Obs}$	$e_{14}e_{13}$	0.01532699	0.01698303	0.90248889	0.97510402
	$e_{13}e_{37}$	0.01619049	0.01698303	0.95333333	
	$e_{37}e_{36}$	0.01698303	0.01698303	1.0	
	$e_{14}e_{13}e_{37}$	0.00199506	0.00221062	0.90248889	
	$e_{13}e_{37}e_{36}$	0.00210746	0.00221062	0.95333333	
$e_{14}e_{13}e_{37}e_{36}$	0.00025955	0.00028759	0.90248889		
$\mathcal{L}_{p_3}^{Obs}$	$\emptyset$	/	/	/	0

By analysing the figure 4.5.6, several point can be noted.

1. As wanted, in this example case, a decision cannot be directly done since the two activities  $\mathcal{A}_1$ : cooking and  $\mathcal{A}_2$ : hot beverage preparation have a too high probability simultaneously. Furthermore, by choosing the most probable one, a mistake can be done since the events used in this example correspond, in real life, to the events generated when an inhabitant put some water to boil. This step, necessary to prepare hot beverages and to prepare pasta, is less closed to represent the activity  $\mathcal{A}_1$  than the activity  $\mathcal{A}_2$  since the other way to cook is by preparing a ready-cooked dishes, which do not need water;
2. The natural property of classical likelihood to decrease with the length of the considered sequence is clearly visible here;
3. we can see that, if the language was not created and the normalised likelihood directly used, the activity  $\mathcal{A}_1$  have a distance value of 0.435 against 0.9025 for the  $\mathcal{A}_2$ . Indeed, due to the influence of the low probability linked to the event  $e_{14}$  in the PFA  $\mathcal{A}_1$ , indeterminism is less visible and a faulty conclusion could be done without the language creation.

By evaluating the distance at each time step, it is possible to draw in a figure the evolution of the probability with the time. A simple example of its kind of figure is given figure 4.5.1 and shows the evolution of this value during a deterministic realisation of the  $\mathcal{A}_1$ : cooking activity. Here, in order to discuss the results, the log of actually performed activities is compared with the computed estimations. We enlighten this log is for the validation procedure only and is not required by the proposed method.

The knowledge of performed activity is drawn with plain lines. The probability is equal to 1 when the activity is actually performed. The value of the presented estimator is drawn by the crossed lines.

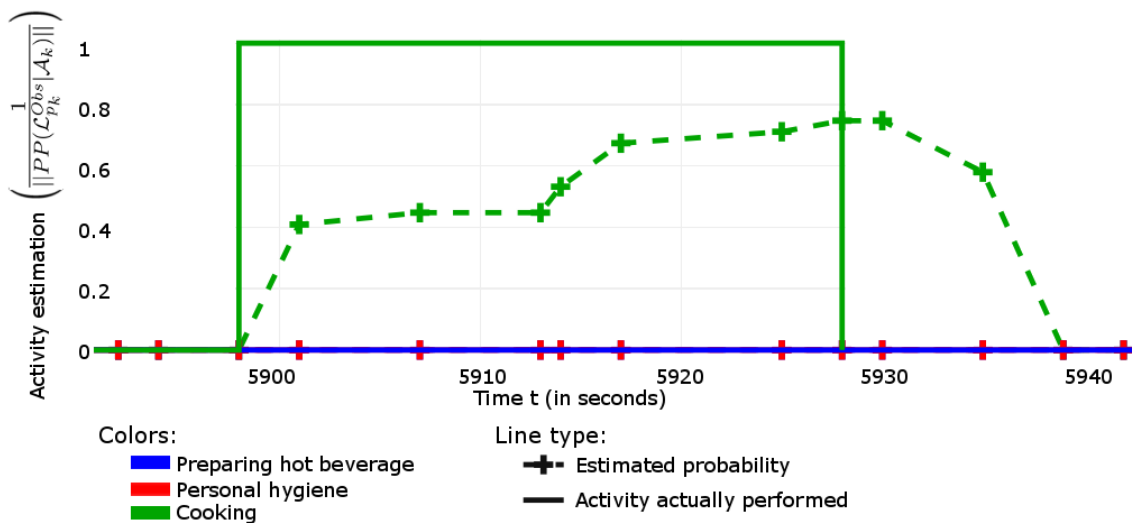


Figure 4.5.1: Inverse of the normalised perplexity during the realisation of the activity  $\mathcal{A}_1$  cooking using no shared events.

The example shows that for each new observed event, the estimation of probability is actualised. The language is empty if the projected sequence length is lower than 2

and an offset is systematically observed when the activity starts. Furthermore, another offset is present when an activity stops. This second offset is due to the use of a sliding window storing the last five observed events.

## 4.6 Discussion

In the previous sections, we present an AR method in four steps and we applied it into the case study presented in 2. The following discussion consists of analysing results obtained depending on the activities of daily living performance. To this aim, two specific cases are considered:

- the recognition of an activity occurring individually,
- the recognition of a succession of activities.

With the first case, presented section 4.6.1, we study the activities to monitor probabilities obtained when an activity is performed after a succession of events linked to no other activities. In this way, we are able to conclude about the AR method without introducing the disturbances possibly linked to an activity violently changed or interrupted.

With the second case, presented section 4.6.2, we study the result of the AR method when the activities are performed successively without interruption. In this way, we see the reaction of our method to an activity change or interruption.

As explained in section 4.5.4, in order to discuss the results, the log of actually performed activities is compared with the computed estimations. We enlighten this log is for the validation procedure only and is not required by the proposed method.

In the figures presented in the next sections, the knowledge of performed activity is drawn with plain lines. The probability is equal to 1 when the activity is actually performed. The value of the presented estimator is drawn by the crossed lines.

### 4.6.1 Activities occurring individually

In this subsection, in order to estimate our method viability, the AR protocol is applied to our three case study activities:  $\mathcal{A}_1$ : cooking,  $\mathcal{A}_2$ : hot beverage preparation and  $\mathcal{A}_3$ : use bathroom. These three activities have the advantage to allow us to test three specific cases:

1.  $w_{\mathcal{A}_3}^{Obs}$  is an activity sharing no event with any other ones,
2. the activity  $w_{\mathcal{A}_2}^{Obs}$  shares some events with the activity  $\mathcal{A}_1$  when the action *prepare coffee* is performed,
3. the activity  $w_{\mathcal{A}_1}^{Obs}$  shares a lot of events with the activity  $\mathcal{A}_2$  when the action *prepare pasta* is performed. Respectively, the activity  $w_{\mathcal{A}_2}^{Obs}$  shares a lot of events with the activity  $\mathcal{A}_1$ .

*1 - Activity sharing no event with any other ones:*

Figure 4.6.1 shows the result of the AR applied to the performance of the activity  $\mathcal{A}_3$ : use bathroom. We observe that the values of the inverse of the normalised perplexity linked to the activity  $\mathcal{A}_3$  is clearly upper than zero since the perplexity linked to the other activities  $\mathcal{A}_1$  and  $\mathcal{A}_2$  are equals to zero.

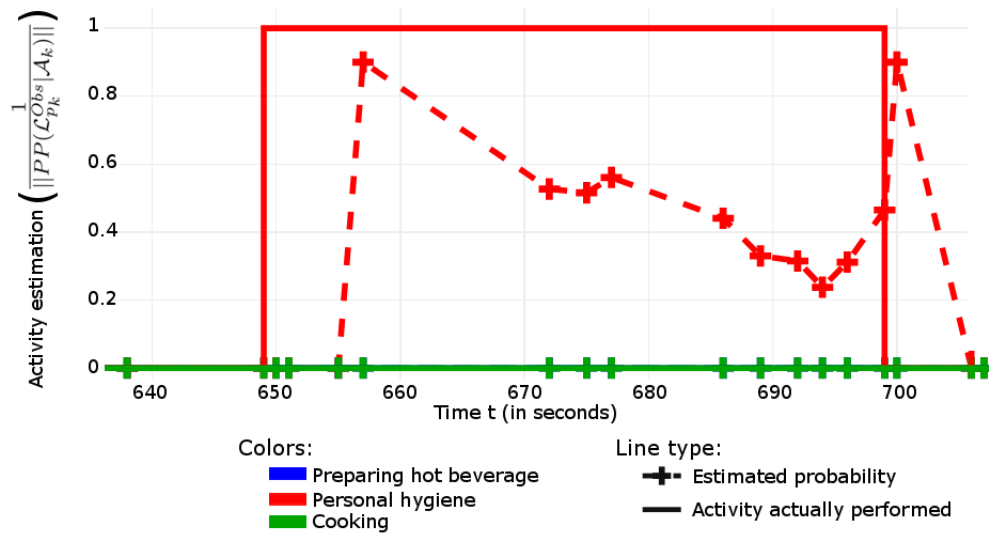


Figure 4.6.1: Inverse of the normalised perplexity during the realisation of the activity  $\mathcal{A}_3$ : *use bathroom* sharing no event with any other activity.

In this case, it seems obvious that the performed activity is  $\mathcal{A}_3$ : *use bathroom* since the obtained distance is clearly good for  $\mathcal{A}_3$  and no other activity seems to be occurring.

Thus, as expected, the proposed AR protocol for activities sharing no event with others is efficient when the concerned activities are occurring individually.

## 2 - Activity sharing some events with another one:

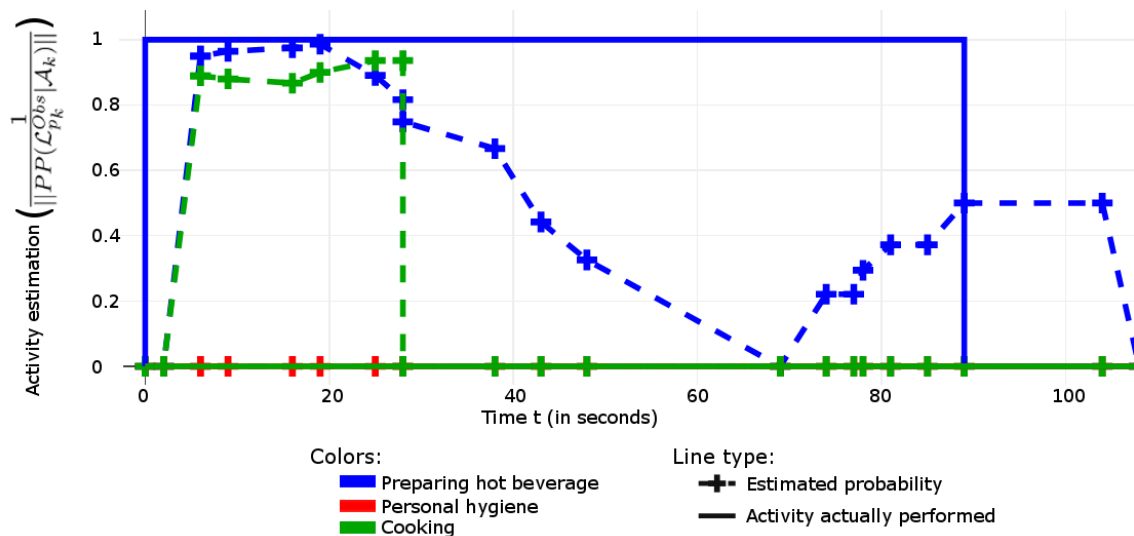


Figure 4.6.2: Inverse of the normalised perplexity during the realisation of the activity  $\mathcal{A}_2$ : *hot beverage preparation* sharing some events with the activity  $\mathcal{A}_1$ : *cooking*.

Figure 4.6.2 shows the result of the AR applied to the performance of the activity  $\mathcal{A}_2$ : *hot beverage preparation*, more specifically during the preparation of coffee. We can observe that the values of the inverse of the normalised perplexity linked to the activity  $\mathcal{A}_1$ : *cooking* and  $\mathcal{A}_2$ : *hot beverage preparation* are initially both clearly upper than



zero since the perplexity linked to the activity  $\mathcal{A}_3$  is equal to zero. Furthermore, before twenty-three seconds, we are in a situation of indeterminism. Then, once unshared event occurs, the indeterminism disappears since the activity  $\mathcal{A}_1$ : *cooking* probability fall to zero and never grown up.

In this case, it seems obvious after a period of indecision that the performed activity is  $\mathcal{A}_2$ : *hot beverage preparation* since the obtained distance is good for  $\mathcal{A}_2$  longer than for  $\mathcal{A}_1$ .

The proposed AR protocol for activities sharing some events with others is efficient once unshared events appear when the concerned activities are occurring individually.

### 3 - Activity sharing many events with another one:

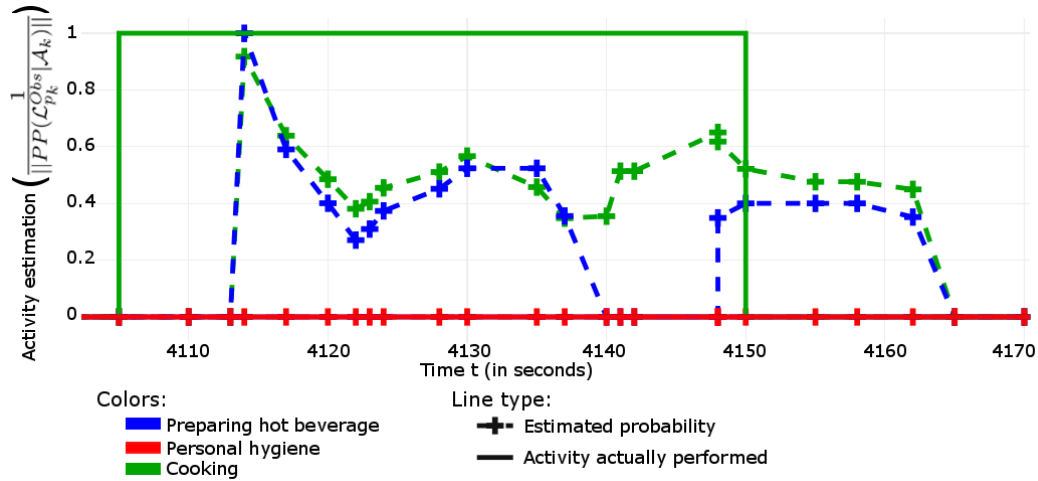


Figure 4.6.3: Inverse of the normalised perplexity during the realisation of the activity  $\mathcal{A}_1$ : *cooking* sharing many events with the activity  $\mathcal{A}_2$ : *hot beverage preparation*.

Figure 4.6.3 shows the result of the AR applied to the performance of the activity  $\mathcal{A}_1$ : *cooking*, more specifically during pasta preparation. We can observe that the values of the inverse of the normalised perplexity linked to the activity  $\mathcal{A}_1$ : *cooking* and  $\mathcal{A}_2$ : *hot beverage preparation* are both clearly upper than zero the majority of the time since the perplexity linked to the activity  $\mathcal{A}_3$  and  $\mathcal{A}_2$  are equals to zero. Here, the situation of indeterminism is too present to clearly conclude about which activity was performed.

The proposed AR protocol for activities sharing many events with others is not efficient. This drawback of the methodology is normal and common to each recognition technique. Of course, if two activities have too closed moves indissociable using sensors installed in the smart home, our AR method could not dissociate them either.

In addition to these three cases, it is important to highlight that if a sensor is linked with too many activities, it is observed a lot of times during the learning period and its probability becomes predominant in all linked ADL models. This property makes the occurrence of the concerned event non-discriminant and noisy. Therefore, the events detected by such sensors are not useful and may be unlinked to all activities. If used for the ADL monitoring only, this sensors should not be implemented.

## 4.6.2 Activities occurring successively

In this subsection, in order to estimate if our method is robust to activity interruption and succession, the AR protocol is applied to a succession of the three case study

activities:  $\mathcal{A}_3$ : use bathroom then  $\mathcal{A}_1$ : cooking and finally  $\mathcal{A}_2$ : hot beverage preparation.

As we see before that AR is not efficient with activities performances sharing many events, the succession tested in this subsection implied only activity occurrences having zero or some events shared.

Figure 4.6.4 the result of our AR method for the presented activity succession. We can observe that it is easy to conclude from the probabilities that we perform a succession of activities  $\mathcal{A}_3$  then  $\mathcal{A}_1$  finally  $\mathcal{A}_2$ . The detection of the exact time of changing activity is more difficult between the two activities  $\mathcal{A}_1$  and  $\mathcal{A}_2$  sharing some events but the succession can even be detected.

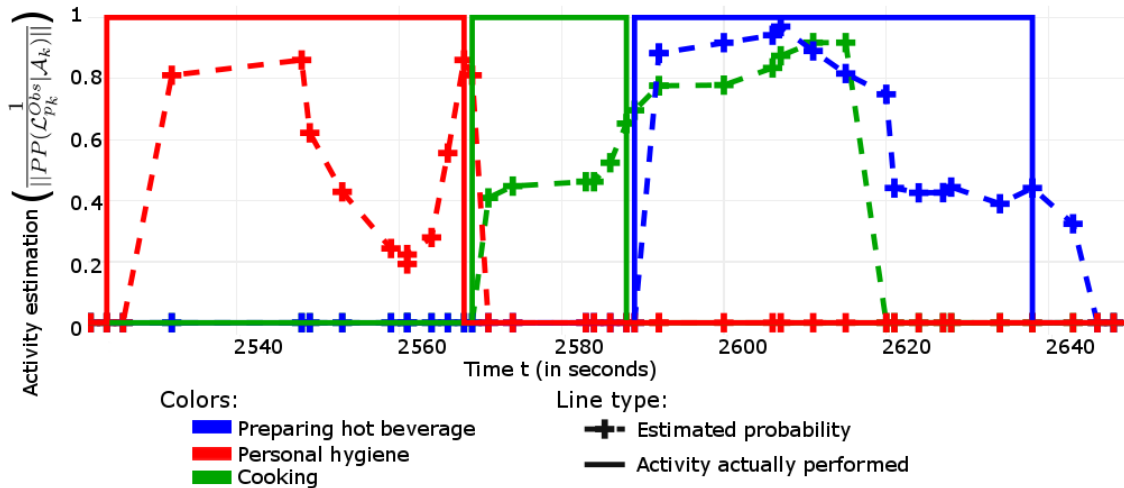


Figure 4.6.4: Inverse of the normalised perplexity during the succession of the three activities  $\mathcal{A}_1$ : cooking,  $\mathcal{A}_2$ : hot beverage preparation and  $\mathcal{A}_3$ : use bathroom sharing no zero of some events.

### 4.6.3 Validation

The presented and discussed results are obtained by discovering activities as described in chapter 3 and by applying the activity recognition protocol presented in this chapter 4.

In this discussion section, we showed that, after learning PFAs, we are able to find the performed activity with the common hypothesis to have activities not sharing too much events.

These results allow us to conclude that the presented method to discover and recognise activities of daily living performed by an inhabitant of a smart home is powerful. Especially, those results are obtained without declaring the performed activity during the learning period and using only environmental and binary sensors.

## Conclusion

An approach for activity recognition has been presented in this chapter. To do that, a new probabilistic distance, called normalised likelihood, and its extension to *perplexity* has been defined. It has also been shown that the normalised likelihood can be efficiently computed onto a reduced model of the activity without loss of accuracy. This make efficient the on-line computation of the probability for an activity, modelled by a

Probabilistic Finite-State Automaton, to be performed. Finally, the proposed approach has been applied to several test activities performed in the living lab presented in chapter 2 and the quality of the results obtained has been discussed.

# Conclusions & Outlooks

---

## Summary

The objective of this thesis was to propose a contribution in the field of ambient assisted living and activity of daily living monitoring by limiting the semantic level of the problem inputs. Indeed, three main limiting issues in the domain appear to be non-negligible: the instrumentation cost, the human non-determinism and the patient inability and privacy. These three points lead us to make four assumptions:

Assumption 1 Activities are represented by probabilistic models;

Assumption 2 Only binary and environmental sensors are used;

Assumption 3 The considered smart home has a single inhabitant;

Assumption 4 The knowledge of the actually performed activity is not required.

These assumptions bring us to use paradigms, theory and tools of the discrete event system domain. By considering an inhabitant generating sensor events by living in a smart home and using an initial expert knowledge, a new framework to discover and recognise activities performed by the inhabitant was proposed. In order to develop this framework, we built a living lab adapted to our hypothesis and we developed two main contributions, detailed and presented in this thesis.

### **A living lab installation and a protocol to test the developed methods**

Since the subject treated in this thesis is directly related to a real problem of society requiring concrete solutions, all the methods presented in this thesis must be applicable on a real case. The development of the methods cannot be decoupled with the technical application. Therefore, in the first time, several living labs were introduced and their utility in applying our methods using their datasets was discussed. Unfortunately, the limitations of the assumptions previously taken make living labs currently known not fully adapted to our methods. These incompatibilities are mainly due to a lack of information on smart homes equipped.

To cope with this problem, a new living laboratory has been developed at ENS Paris-Saclay. This ambient assisted living test area (AALTA) is presented and the placement of objects and sensors is given. A necessary expert decomposition in the activity discovery method is also detailed. Finally, the experimental protocol used to generate test databases is developed and explained.

### **An activity discovery method using probabilistic finite-state automata**

An approach to discover the human activities of daily living has been proposed and illustrated. An activity modelling procedure using probabilistic finite-state automata was developed based on the knowledge of a training event log database and thanks to an expert decomposition of activities into actions related to the smart home events. The main advantage of this method is related to the Assumption 4 of this thesis which avoids the use of activity labelling during the learning phase to generate the models.

This activity discovery method has been illustrated using the living lab information.

### **An activity recognition method using previously generated models**

Finally, an activity recognition approach has been presented. This approach, decomposable into four steps, needs a new probabilistic distance, called *normalised likelihood*, and its extension to the *normalised perplexity*. These two distance estimators are defined and explained in this thesis. It has also been shown that the *normalised likelihood* can be calculated efficiently using a reduced model without loss of precision. This makes efficient the on-line calculation of the probabilities for an activity to be performing. Finally, the recognition approach has been applied to several test activities carried out in the living lab previously presented and the quality of the results obtained appear to be good.

## **Outlook**

A global framework for discovering and recognising activities is proposed in this thesis and several improvements can be envisaged.

### **To relax the assumption concerning the sensor technology used**

In this thesis, in addition to the rejection of too intrusive sensors (as cameras), we reject in Assumption 2 the use of wearable sensors sometimes incompatible with some pathologies. This assumption leads to consider only single inhabitant smart homes. A perspective is to relax this assumption, by authorising, when possible, the use of wearable sensors which are not too intrusive considering the inhabitant privacy: the binary ones (as RFID). By allowing the use of binary and wearable sensors, the assumption of single inhabitant can easily be removed since the automatic labelling of the input data with the name of the person wearing sensors is possible. Furthermore, the use of RFID targets fixed on some objects can lead to a better granularity in the expert knowledge. The presented method, developed to deal with binary sensors, should directly be applicable without changes if the Assumption 2 is relaxed by allowing wearable binary sensors.

### **To use the discovered models and the recognised activity to treat the detection of deviation and activity prediction problems**

As a continuity of the presented work, it can be envisaged to treat the two other main objectives linked with the activity of daily living monitoring: the detection of deviations and the activity prediction. Indeed, as we did with the activity discovery and recognition, those two other objectives could be reformulated to be compatible with the discrete event system paradigms.

For example, the use of well-known diagnosis methods of the industrial domain can be extended to human deviation if we consider human deviations as faults to detect. Since, in this thesis, the human behaviour are models using DES paradigms, these extensions can easily be envisaged. An extension of residuals uses for fault localisation developed in the LURPA (Roth et al., 2011) can be extended to probabilistic residuals for human deviation detection.

Furthermore, works treating the field of predictability in DES domain such as Jérón et al. (2008) and Genc and Lafortune (2009) can be reused and adapted to our models.

---

### **To identify activities not listed by the medical staff**

Using the existing methods in the industrial identification of reactive system (Dotoli et al., 2008; Saives et al., 2015), it could be envisaged to find activities initially not given by the medical staff, to detect the linked events and to automatically discover their models without any expert intervention.

The medical staff and the expert decomposition could thus be enriched by automatically detected activities one which deviation detection could also be performed. The use of identification method to extend the monitored activities to medically not targeted ones could allow a better inhabitant supervision during his entire daily life.



# Bibliography

---

- Agarwal, Y., Balaji, B., Gupta, R., Lyles, J., Wei, M., and Weng, T. (2010). Occupancy-driven energy management for smart building automation. In *Proceedings of the 2nd ACM workshop on embedded sensing systems for energy-efficiency in building*, pages 1–6. ACM.
- Agrawal, R. and Srikant, R. (1995). Mining sequential patterns. In *Data Engineering, 1995. Proceedings of the Eleventh International Conference on*, pages 3–14. IEEE.
- Ahad, M. A. R., Tan, J., Kim, H., and Ishikawa, S. (2008). Human activity recognition: Various paradigms. In *Control, Automation and Systems, 2008. ICCAS 2008. International Conference on*, pages 1896–1901. IEEE.
- Andriluka, M., Roth, S., and Schiele, B. (2009). Pictorial structures revisited: People detection and articulated pose estimation. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1014–1021. IEEE.
- Balta-Ozkan, N., Boteler, B., and Amerighi, O. (2014). European smart home market development: Public views on technical and economic aspects across the united kingdom, germany and italy. *Energy Research & Social Science*, 3:65–77.
- Barlow, J. and Venables, T. (2003). Smart home, dumb suppliers? the future of smart homes markets. In *Inside the Smart Home*, pages 247–262. Springer.
- Baum, L. E. (1972). An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. In Shisha, O., editor, *Inequalities III: Proceedings of the Third Symposium on Inequalities*, pages 1–8, University of California, Los Angeles. Academic Press.
- Bellifemine, F., Fortino, G., Giannantonio, R., Gravina, R., Guerrieri, A., and Sgroi, M. (2011). Spine: a domain-specific framework for rapid prototyping of wbsn applications. *Software: Practice and Experience*, 41(3):237–265.
- Brdiczka, O., Crowley, J. L., and Reignier, P. (2009). Learning situation models in a smart home. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(1):56–63.
- Cassandras, C. G. and Lafortune, S. (2009). *Introduction to discrete event systems*. Springer Science & Business Media.
- Charaoui, A. A., Climent-Pérez, P., and Flórez-Revuelta, F. (2012). A review on vision techniques applied to human behaviour analysis for ambient-assisted living. *Expert Systems with Applications*, 39(12):10873–10888.



- Chan, M., Estève, D., Escriba, C., and Campo, E. (2008). A review of smart homes—present state and future challenges. *Computer methods and programs in biomedicine*, 91(1):55–81.
- Chandola, V., Banerjee, A., and Kumar, V. (2012). Anomaly detection for discrete sequences: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 24(5):823–839.
- Chikhaoui, B., Wang, S., and Pigot, H. (2010). A new algorithm based on sequential pattern mining for person identification in ubiquitous environments. In *KDD workshop on knowledge discovery from sensor data*, pages 19–28.
- Chikhaoui, B., Wang, S., and Pigot, H. (2011). A frequent pattern mining approach for ads recognition in smart environments. In *Advanced Information Networking and Applications (AINA), 2011 IEEE International Conference on*, pages 248–255. IEEE.
- Cook, D. J. and Krishnan, N. C. (2015). *Activity Learning: Discovering, Recognizing, and Predicting Human Behavior from Sensor Data*. John Wiley & Sons.
- Cook, D. J., Krishnan, N. C., and Rashidi, P. (2013). Activity discovery and activity recognition: A new partnership. *IEEE transactions on cybernetics*, 43(3):820–828.
- Cook, D. J., Schmitter-Edgecombe, M., et al. (2009). Assessing the quality of activities in a smart environment. *Methods of information in medicine*, 48(5):480.
- Cornelis, E., Gorus, E., Beyer, I., Bautmans, I., and De Vriendt, P. (2017). Early diagnosis of mild cognitive impairment and mild dementia through basic and instrumental activities of daily living: Development of a new evaluation tool. *PLoS medicine*, 14(3):e1002250.
- Cover, T. M. and Thomas, J. A. (1991). *Elements of information theory*.
- Darianian, M. and Michael, M. P. (2008). Smart home mobile rfid-based internet-of-things systems and services. In *Advanced Computer Theory and Engineering, 2008. ICACTE'08. International Conference on*, pages 116–120. IEEE.
- Devaraju, A. and Hoh, S. (2008). Ontology-based context modeling for user-centered context-aware services platform. In *Information Technology, 2008. ITSIM 2008. International Symposium on*, volume 2, pages 1–7. IEEE.
- Dotoli, M., Fanti, M. P., and Mangini, A. M. (2008). Real time identification of discrete event systems using petri nets. *Automatica*, 44(5):1209–1219.
- Duong, T., Phung, D., Bui, H., and Venkatesh, S. (2009). Efficient duration and hierarchical modeling for human activity recognition. *Artificial intelligence*, 173(7-8):830–856.
- Duong, T. V., Bui, H. H., Phung, D. Q., and Venkatesh, S. (2005). Activity recognition and abnormality detection with the switching hidden semi-markov model. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 838–845. IEEE.
- EUROSTAT, E. (2010). *Population structure and ageing*.

- EUROSTAT, E. (2013). Population structure and ageing.
- Farlex (2018). activities of daily living definition.
- Fleury, A., Vacher, M., and Noury, N. (2010). Svm-based multimodal classification of activities of daily living in health smart homes: sensors, algorithms, and first experimental results. *Information Technology in Biomedicine, IEEE Transactions on*, 14(2):274–283.
- Flöck, M. (2010). *Activity monitoring and automatic alarm generation in AAL-enabled homes*. Logos Verlag Berlin GmbH.
- Forkan, A., Khalil, I., and Tari, Z. (2014). Cocamaal: A cloud-oriented context-aware middleware in ambient assisted living. *Future Generation Computer Systems*, 35:114–127.
- Forkan, A. R. M., Khalil, I., Tari, Z., Fougou, S., and Bouras, A. (2015). A context-aware approach for long-term behavioural change detection and abnormality prediction in ambient assisted living. *Pattern Recognition*, 48(3):628–641.
- Gaglio, S., Re, G. L., and Morana, M. (2015). Human activity recognition process using 3-d posture data. *IEEE Transactions on Human-Machine Systems*, 45(5):586–597.
- Genc, S. and Lafortune, S. (2009). Predictability of event occurrences in partially-observed discrete-event systems. *Automatica*, 45(2):301–311.
- Himmel, S., Ziefle, M., and Arning, K. (2013). *From Living Space to Urban Quarter: Acceptance of ICT Monitoring Solutions in an Ageing Society*, pages 49–58. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Hussain, S., Schaffner, S., and Moseychuck, D. (2009). Applications of wireless sensor networks and rfid in a smart home environment. In *Communication Networks and Services Research Conference, 2009. CNSR'09. Seventh Annual*, pages 153–157. IEEE.
- Intille, S. S., Tapia, E. M., Rondoni, J., Beaudin, J., Kukla, C., Agarwal, S., Bao, L., and Larson, K. (2003). Tools for studying behavior and technology in natural settings. In *UbiComp 2003: Ubiquitous Computing*, pages 157–174. Springer.
- Jelinek, F. (1997). *Statistical methods for speech recognition*. MIT press.
- Jéron, T., Marchand, H., Genc, S., and Lafortune, S. (2008). Predictability of sequence patterns in discrete event systems. *IFAC Proceedings Volumes*, 41(2):537–543.
- Jiang, L., Liu, D.-Y., and Yang, B. (2004). Smart home research. In *Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference on*, volume 2, pages 659–663. IEEE.
- Junker, H., Amft, O., Lukowicz, P., and Tröster, G. (2008). Gesture spotting with body-worn inertial sensors to detect user activities. *Pattern Recognition*, 41(6):2010–2024.
- Jurafsky, D. and Martin, J. H. (2014). *Speech and language processing*, volume 3. Pearson London.
- Kellokumpu, V., Pietikäinen, M., and Heikkilä, J. (2005). Human activity recognition using sequences of postures. In *MVA*, pages 570–573.

- Kim, E., Helal, S., and Cook, D. (2010). Human activity recognition and pattern discovery. *Pervasive Computing, IEEE*, 9(1):48–53.
- King, N. (2003). Smart home—a definition. *Intertek Research and Testing Center*, pages 1–6.
- Kleinberger, T., Becker, M., Ras, E., Holzinger, A., and Müller, P. (2007). *Ambient Intelligence in Assisted Living: Enable Elderly People to Handle Future Interfaces*, pages 103–112. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Krishnan, N. C. and Cook, D. J. (2014). Activity recognition on streaming sensor data. *Pervasive and Mobile Computing*, 10, Part B:138 – 154.
- Krishnan, N. C. and Panchanathan, S. (2008). Analysis of low resolution accelerometer data for continuous human activity recognition. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pages 3337–3340. IEEE.
- Krumm, J. and Horvitz, E. (2006). Predestination: Inferring destinations from partial trajectories. In *International Conference on Ubiquitous Computing*, pages 243–260. Springer.
- Lara, O. D. and Labrador, M. A. (2013). A survey on human activity recognition using wearable sensors. *Communications Surveys & Tutorials, IEEE*, 15(3):1192–1209.
- Lawton, M. P. and Brody, E. M. (1969). Assessment of older people: self-maintaining and instrumental activities of daily living. *The gerontologist*, 9(3\_Part\_1):179–186.
- Li, W., Zhang, Z., and Liu, Z. (2008). Expandable data-driven graphical modeling of human actions based on salient postures. *IEEE transactions on Circuits and Systems for Video Technology*, 18(11):1499–1510.
- Mahmoud, S., Lotfi, A., and Langensiepen, C. (2013). Behavioural pattern identification and prediction in intelligent environments. *Applied Soft Computing*, 13(4):1813–1822.
- Mihailidis, A., Carmichael, B., and Boger, J. (2004). The use of computer vision in an intelligent environment to support aging-in-place, safety, and independence in the home. *IEEE Transactions on information technology in biomedicine*, 8(3):238–247.
- Mocholí, J., Sala, P., Fernández-Llatas, C., and Naranjo, J. (2010). Ontology for modeling interaction in ambient assisted living environments. In *XII Mediterranean Conference on Medical and Biological Engineering and Computing 2010*, pages 655–658. Springer.
- Monekosso, D., Florez-Revuelta, F., and Remagnino, P. (2015). Ambient assisted living [guest editors’ introduction]. *IEEE Intelligent Systems*, 30(4):2–6.
- Ney, H. (1997). Corpus-based statistical methods in speech and language processing. In *Corpus-based methods in language and speech processing*, pages 1–26. Springer.
- Norris, T., Clarke, T. C., and Schiller, J. S. (2017). Early release of selected estimates based on data from the january–march 2017 national health interview survey.

- Oliver, N., Horvitz, E., and Garg, A. (2002). Layered representations for human activity recognition. In *Proceedings of the 4th IEEE International Conference on Multimodal Interfaces*, page 3. IEEE Computer Society.
- Otto, C., Milenkovic, A., Sanders, C., and Jovanov, E. (2006). System architecture of a wireless body area sensor network for ubiquitous health monitoring. *Journal of mobile multimedia*, 1(4):307–326.
- Park, S. and Kautz, H. (2008). Hierarchical recognition of activities of daily living using multi-scale, multi-perspective vision and rfid.
- Rabiner, L. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.
- Reisberg, B., Finkel, S., Overall, J., Schmidt-Gollas, N., Kanowski, S., Lehfeld, H., Hulla, F., Sclan, S. G., Wilms, H.-U., Heininger, K., et al. (2001). The alzheimer’s disease activities of daily living international scale (adl-is). *International Psychogeriatrics*, 13(2):163–181.
- Robles, R. J. and Kim, T.-h. (2010). Applications, systems and methods in smart home technology: A. *Int. Journal of Advanced Science And Technology*, 15.
- Roth, M., Lesage, J.-J., and Litz, L. (2011). The concept of residuals for fault localization in discrete event systems. *Control Engineering Practice*, 19(9):978–988.
- Rouse, M. and Wigmore, I. (2017). Defintion: ubiquitous networking.
- Sadek, I., Biswas, J., Fook, V. F. S., and Mokhtari, M. (2015). Automatic heart rate detection from fbg sensors using sensor fusion and enhanced empirical mode decomposition. In *Signal Processing and Information Technology (ISSPIT), 2015 IEEE International Symposium on*, pages 349–353. IEEE.
- Saives, J., Pianon, C., and Faraut, G. (2015). Activity discovery and detection of behavioral deviations of an inhabitant from binary sensors. *Automation Science and Engineering, IEEE Transactions on*, 12(4):1211–1224.
- Sapp, B., Toshev, A., and Taskar, B. (2010). Cascaded models for articulated pose estimation. In *European conference on computer vision*, pages 406–420. Springer.
- Schmitter-Edgecombe, M., Woo, E., and Greeley, D. R. (2009). Characterizing multiple memory deficits and their relation to everyday functioning in individuals with mild cognitive impairment. *Neuropsychology*, 23(2):168.
- Scott, F. (2007). *Teaching homes to be green: smart homes and the environment*. Green Alliance.
- Shah, R. C. and Rabaey, J. M. (2002). Energy aware routing for low energy ad hoc sensor networks. In *Wireless Communications and Networking Conference, 2002. WCNC2002. 2002 IEEE*, volume 1, pages 350–355. IEEE.
- Tapia, E. M., Intille, S. S., and Larson, K. (2004). *Activity recognition in the home using simple and ubiquitous sensors*. Springer.

- Taylor, A. S., Harper, R., Swan, L., Izadi, S., Sellen, A., and Perry, M. (2007). Homes that make us smart. *Personal and Ubiquitous Computing*, 11(5):383–393.
- United Nations (2017). World population prospects, the 2017 revision. *Working Paper*.
- Van Kasteren, T., Englebienne, G., and Kröse, B. J. (2010). An activity monitoring system for elderly care using generative and discriminative models. *Personal and ubiquitous computing*, 14(6):489–498.
- Van Kasteren, T., Noulas, A., Englebienne, G., and Kröse, B. (2008). Accurate activity recognition in a home setting. In *Proceedings of the 10th international conference on Ubiquitous computing*, pages 1–9. ACM.
- Viard, K., Fanti, M. P., Faraut, G., and Lesage, J.-J. (2017). Recognition of Human Activity Based on Probabilistic Finite-State Automata. In *22nd IEEE International Conference on Emerging Technologies And Factory Automation*, Limassol, Cyprus.
- Vidal, E., Thollard, F., De La Higuera, C., Casacuberta, F., and Carrasco, R. C. (2005a). Probabilistic finite-state machines - Part I. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(7):1013–1025. 12.
- Vidal, E., Thollard, F., De La Higuera, C., Casacuberta, F., and Carrasco, R. C. (2005b). Probabilistic finite-state machines - Part II. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(7):1026–1039.
- World Health Organization (2012). Good health adds life to years: Global brief for world health day 2012.
- Wu, B. and Nevatia, R. (2005). Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 90–97. IEEE.
- Zhou, Z., Chen, X., Chung, Y.-C., He, Z., Han, T. X., and Keller, J. M. (2008). Activity analysis, summarization, and visualization for indoor human activity monitoring. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(11):1489–1498.

# Résumé en langue française

## Introduction

Au cours des dernières décennies, la proportion de personnes âgées de 65 ans et plus dans le monde a continuellement augmenté. Comme estimé dans l'étude statistique européenne [EUROSTAT \(2013\)](#) et explicité par la figure [Fr.1](#), ce vieillissement de la population devrait continuer dans le futur. Or, comme montré par l'étude américaine [Norris et al. \(2017\)](#) et représenté sur la figure [Fr.2](#), les problèmes de santé viennent avec l'âge et 55% des personnes âgées ont des problèmes de santé. D'après la même étude, seulement 8% des personnes âgées de 65 ans et plus ont besoin qu'une tierce personne les aide au quotidien, ce qui laisse 47% de personnes de cette tranche d'âge avec des problèmes de santé suffisamment peu sérieux pour leur permettre de vivre seuls.

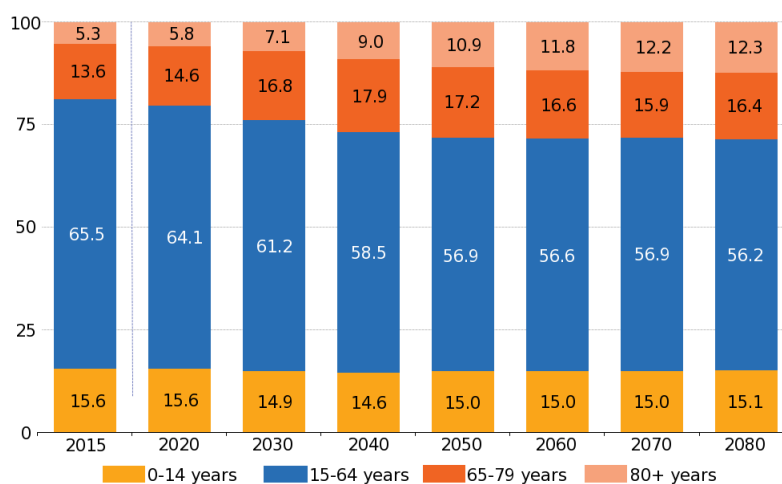


Figure Fr.1: Structure de la population par tranche d'âge, UE-28, 2015-2080 (en % de la population totale) ([EUROSTAT, 2013](#)).

Cette évolution démographique de la société soulève de nouveaux problèmes, tant sur le plan humain qu'économique. En effet, les institutions de santé ne sont pas capables, en l'état actuel, de gérer autant de personnes âgées. C'est pourquoi il est nécessaire de développer des solutions alternatives aux moyens existants afin d'aider les personnes âgées n'ayant pas de problèmes de santé trop graves à rester indépendantes des services de santé conventionnels.

Une solution actuellement envisagée est de développer des systèmes assistants et surveillant les personnes continuellement afin de leur permettre de rester à domicile aussi longtemps que possible. Ainsi, les installations médicales peuvent être réservées aux pathologies sévères et aux urgences. De plus, cette solution permet aux personnes âgées ou handicapées de rester plus longtemps dans un environnement qui leur est familier

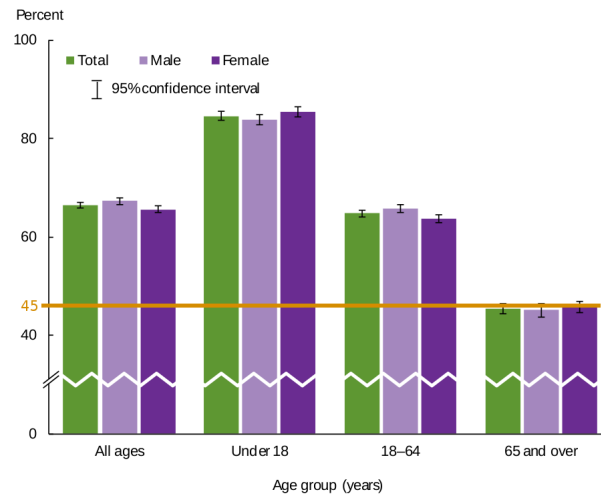


Figure Fr.2: Pourcentage de la population de tout âge ayant un excellent ou un très bon état de santé, par âge et par sexe : États-Unis (Norris et al., 2017).

(Cornelis et al., 2017). C'est pourquoi cette solution, moins chère et plus plaisante, doit être développée. Cette surveillance continue des personnes à domicile permettrait de contacter la famille ou des médecins quand un problème ou un comportement anormal est détecté.

La surveillance des activités de tous les jours (ADL) est l'une des principales possibilités étudiées dans le domaine des systèmes de santé à domicile. Elle consiste à donner au personnel médical des informations précises et utiles à propos du patient. Par définition (Lawton and Brody, 1969), une ADL est une activité qui est réalisée régulièrement par une personne (comme préparer à manger, faire le ménage, prendre du temps libre, etc.) et dont la surveillance est utile aux médecins. Les études actuelles sur les ADLs traitent quatre points: la découverte d'activité (AD), la reconnaissance d'activité (AR), la prédiction d'activité (AP) et la détection de déviation de comportement (DD). Dans cette thèse, seulement les problèmes de découverte et de reconnaissance d'activité sont traités. L'objectif de l'AD est de générer un ou plusieurs modèles d'activités plus ou moins formels en observant, durant une période d'apprentissage, le comportement de la personne à surveiller. L'objectif de l'AR est de détecter qu'une activité est réalisée quand la personne la réalise effectivement.

L'objectif de cette thèse est de proposer une méthode de découverte d'activité et une méthode de reconnaissance d'activité compatible avec les limitations choisies suivantes:

1. Les activités sont représentées par des modèles probabilistes : ces modèles sont compatibles avec le côté erratique du comportement humain.
2. Seulement des capteurs binaires et environnementaux sont utilisés : ces capteurs sont considérés comme non intrusifs et peu chers. L'objectif est donc de savoir jusqu'où on peut aller en n'utilisant que ces capteurs qui sont sémantiquement pauvres.
3. Les maisons équipées considérées pour appliquer notre méthode sont habitées par une seule personne. En effet, l'utilisation de capteurs binaires environnementaux ne permet pas de distinguer précisément quelle personne interagit avec quel capteur. Cette limitation à une seule personne surveillée n'est pas un problème, car,

en pratique, les cibles principales de la surveillance de personnes à domicile sont les personnes vivant seules.

4. Finalement, l'étiquetage des données d'apprentissages avec les activités réellement réalisées est rejeté. En effet, qu'elle soit faite par un expert analysant les données capteurs ou directement par l'habitant pendant la période d'essai, la déclaration manuelle des activités réalisées pendant la période d'apprentissage contient souvent des informations mauvaises ou incomplètes. Afin de parer la perte d'information liée à ce choix, un nouveau savoir est ajouté : la décomposition des activités à surveiller en actions liées à des données capteurs.

Le plan proposer dans cette thèse pour découvrir et reconnaître les activités de tous les jours est donné par la figure Fr.3.

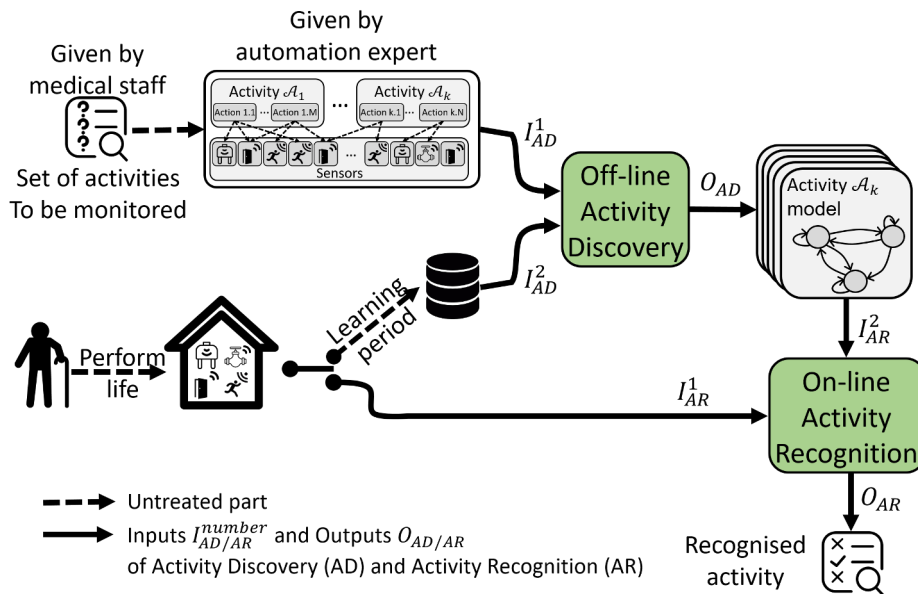


Figure Fr.3: Plan proposé pour la découverte et la reconnaissance d'activités.

## Création d'un appartement test

Comme le sujet traité dans cette thèse est directement lié au problème sociétal présenté précédemment, les méthodes présentées doivent être applicables à des cas réels. En effet, le développement des méthodes ne peut être découplé de l'application technique. Malheureusement, les appartements tests existants dans la littérature ne sont pas compatibles avec nos hypothèses de travail. C'est pourquoi, pour tester les méthodes de cette thèse, un appartement prêté par l'ENS Paris-Saclay a été instrumenté en utilisant vingt capteurs binaires. L'instrumentation a été choisie afin de découvrir et reconnaître trois activités :

- $\mathcal{A}_1$  : Cuisiner;
- $\mathcal{A}_2$  : Préparer une boisson chaude;
- $\mathcal{A}_3$  : Utiliser la salle de bain;

Les activités à découvrir et reconnaître peuvent être décomposées hiérarchiquement en plusieurs actions rattachables à certains capteurs. Cette décomposition sera utilisée



plus tard pour découvrir les modèles d'activités. Dans notre exemple, la décomposition de chaque activité en deux actions et leurs liens avec les événements sont choisis afin de pouvoir analyser les résultats obtenus pour quatre cas spécifiques:

1. Une des activités doit pouvoir être réalisée dans une pièce de l'appartement équipé dans laquelle aucune autre activité ne peut être réalisée. Ainsi, les événements observés dans cette zone seront systématiquement liés à cette activité uniquement. Dans notre cas d'application, l'activité  $\mathcal{A}_3$  répond à ce critère;
2. Une activité doit avoir deux actions sémantiquement proches, mais en pratique, réalisée différemment. Les actions *Préparer un plat préparé* et *Faire des pâtes* de notre cas d'étude répond à ce critère;
3. Un des moyens pour réaliser une activité doit être si proche d'une autre activité que la distinction entre ces deux activités est difficile. Dans notre cas, les actions *préparées des pâtes* de l'activité  $\mathcal{A}_1$  et *préparer du thé* de l'activité  $\mathcal{A}_2$  sont suffisamment proches pour répondre à ce cas;
4. En contraste avec le cas précédent, deux actions d'activités différentes doivent avoir une petite partie de leurs réalisations en commun et une grande partie différente. C'est le cas des actions *préparer des pâtes* de l'activité  $\mathcal{A}_1$  et *préparer du café* de l'activité  $\mathcal{A}_2$ .

## Découverte d'activité

La première contribution de cette thèse est le développement d'une nouvelle méthode de découverte d'activité (AD). Cette approche est nécessaire, car la quatrième limitation rejetant l'étiquetage des données d'apprentissage présentées précédemment est incompatible avec les méthodes de découverte déjà existantes.

Le principal avantage de la méthode développée est sa portabilité. En effet, elle est applicable dans tous les appartements équipés, quelque soit la pathologie de l'habitant.

Cette approche modélise chaque activité à surveiller par un automate à état probabiliste (PFA). La perte d'information liée au rejet du savoir des activités réalisées pendant la période d'apprentissage est compensée par l'ajout d'un savoir expert spécifique donnant la décomposition hiérarchique des activités en actions puis en événements capteurs.

Le modèle de chaque activité est généré en trois étapes:

1. La structure du modèle est automatiquement créée à partir de la décomposition experte;
2. La base de données d'apprentissage est analysée en faisant glisser une fenêtre d'observation composée d'un nombre fixe d'événements et des indicateurs de fréquences pertinentes sont calculés;
3. Les probabilités de nos modèles sont calculées en utilisant les indicateurs de fréquence calculés à l'étape 2.

Les modèles générés par cette découverte d'activités sont ensuite utilisés comment entrée pour la reconnaissance d'activité.

## Reconnaissance d'activité

La seconde contribution principale de cette thèse est le développement d'une nouvelle méthode de reconnaissance d'activité (AR). Cette approche consiste à détecter l'activité réalisée par un habitant surveillé pendant qu'il la réalise. Pour cela, le groupe {habitant + maison équipée} est considéré comme un générateur d'événements et les activités à reconnaître sont modélisées par des PFAs. Dans la littérature, il existe plusieurs façons de reconnaître quel PFA a le plus probablement généré une séquence si les différents PFAs comparés sont liés aux mêmes événements. Cependant, si ceux-ci ne partagent pas les mêmes événements, les méthodes sont inutilisables. Or, les modèles que nous avons découverts précédemment ne partagent pas les mêmes événements, car les activités ne sont pas systématiquement liées aux mêmes capteurs. C'est pourquoi la nouvelle méthode de reconnaissance d'activité présentée dans cette thèse est nécessaire.

Un point commun entre les méthodes existantes (Van Kasteren et al., 2008; Kellokumpu et al., 2005) et la nôtre est la définition et l'utilisation d'une *distance*. En effet, dans ces méthodes comme dans la nôtre, le calcul d'une distance est utilisé pour estimer quelle activité a le plus de chance d'être réalisée au vu des observations.

Dans le but d'améliorer les méthodes de reconnaissance existantes et de les adapter à notre cas, plusieurs opérations telles que l'utilisation de fonctions de projections et la création de langage doivent être considérées. La projection aide à supprimer le bruit lié à d'autres activités ou à des capteurs non pertinents en ne gardant que les événements utiles à la détection de chaque activité. La génération d'un langage nous permet de considérer les événements observés comme une succession de plusieurs groupes d'événements plus ou moins indépendants.

Les distances existantes dans la littérature doivent être adaptées pour donner des résultats exploitables après les différentes projections et créations de langage. C'est pourquoi, dans cette thèse, une nouvelle distance : la vraisemblance normalisée, basée sur la définition de la perplexité usuellement utilisée pour calculer la distance entre un modèle et un langage. Le calcul de cette nouvelle distance posant des problèmes de complexité calculatoire, deux algorithmes ont été développés pour rendre ce calcul faisable dans un temps acceptable (moins d'une seconde).

## Conclusion

Dans cette thèse, une approche globale pour découvrir et reconnaître les activités de tous les jours d'un habitant dans une maison équipée est proposée. Ainsi, une procédure pour modéliser les activités par des automates finis probabilistes est développée en utilisant l'enregistrement des événements générés par l'habitant pendant une période d'essai et la décomposition hiérarchique des activités à modéliser en actions liées aux différents événements de capteurs. Puis, une méthode de découverte d'activité basée sur une nouvelle distance appelée perplexité normalisée est présentée. De plus, il est prouvé que cette nouvelle distance peut être efficacement calculée sans aucune perte de performance en utilisant des algorithmes développés dans cette thèse.

Finalement, toutes ces méthodes sont appliquées sur un vrai appartement de test équipé.

Pour prolonger ces travaux, il pourrait être envisagé de traiter le cas de plusieurs habitants vivant dans une maison équipée. Cela est envisageable en relaxant l'hypothèse de n'utiliser que des capteurs binaires environnementaux (en utilisant des capteurs RFID par exemple).

L'utilisation des modèles générés et des activités reconnues peut être envisagée afin de détecter de potentielles déviations d'habitudes de l'habitant pouvant être symptomatique de certaines pathologies.

Enfin, il peut être envisagé d'identifier automatiquement des activités non listées comme "à surveiller" par le corps médical afin d'améliorer notre reconnaissance d'activité. En effet, l'ajout automatique de ce genre d'activité non sensible peut faciliter la reconnaissance en évitant de potentiels faux positifs.

# Appendix A Proofs

---

## Notations

In order to prove Proposition 4.1, the following notation is defined:

- $e_i \in \Sigma_{\mathcal{A}_k}$ ;
- $e_{eq} \in \Sigma_{\mathcal{A}_k}$  such as  $e_{eq} \in \bigcap_{q_l, q_m \in Q_{\mathcal{A}_k}} Equ_{q_l, q_m}(e_i)$ ;
- $\{e_{eq}\}_{e_i}$  is the set of all possible  $e_{eq}$  associated to  $e_i$ ;
- $e_{sup} \in \Sigma_{\mathcal{A}_k}$  such as  $e_{sup} \in \bigcap_{q_l, q_m \in Q_{\mathcal{A}_k}} Geq_{q_l, q_m}(e_i)$  and  $e_{sup} \notin \bigcap_{q_l, q_m \in Q_{\mathcal{A}_k}} Equ_{q_l, q_m}(e_i)$ ;
- $\{e_{sup}\}_{e_i}$  is the set of all possible  $e_{sup}$  associated to  $e_i$ .

Thus, by definition it holds:

$$\{e_{eq}\}_{e_i} + \{e_{sup}\}_{e_i} = \bigcap_{q_l, q_m \in Q_{\mathcal{A}_k}} Geq_{q_l, q_m}(e_i).$$

### Proposition 4.1 (Chapter 4, Page 89)

Let  $\mathcal{A}_k = \langle Q_{\mathcal{A}_k}, \Sigma_{\mathcal{A}_k}, \delta_{\mathcal{A}_k}, I_{\mathcal{A}_k}, F_{\mathcal{A}_k}, P_{\mathcal{A}_k} \rangle$  be a PFA and  $\mathcal{A}_k^r = \langle Q_{\mathcal{A}_k}, \Sigma_{\mathcal{A}_k}^r, \delta_{\mathcal{A}_k}^r, I_{\mathcal{A}_k}, F_{\mathcal{A}_k}, P_{\mathcal{A}_k}^r \rangle$  be the reduced PFA obtained by the reduction procedure  $\mathcal{A}_k \rightarrow \mathcal{A}_k^r$ . Then  $\forall w \in \Sigma_{\mathcal{A}_k}^*$  of length  $|w|$  it holds:

$$\max_{u \in \Sigma_{\mathcal{A}_k^r}^{|w|}} [P(u|\mathcal{A}_k)] = \max_{v \in \Sigma_{\mathcal{A}_k}^{|w|}} [P(v|\mathcal{A}_k)].$$

*Proof.* In equation (4.26), only event  $e_i$  with  $\{e_{sup}\}_{e_i} = \emptyset$  are kept in the reduced PFA  $\mathcal{A}_k^r$ . We prove that the rejection of events having  $\{e_{sup}\}_{e_i} \neq \emptyset$  does not change the value of the maximum likelihood.

Let  $w = w'_1 \dots w'_k \dots e_i \dots w'_{|w|-1} w'_{|w|}$  be a sequence of events and  $e_i$  is one of the events in the sequence.

Let  $w = w'_1 \dots w'_k \dots e_{sup} \dots w'_{|w|-1} w'_{|w|}$  be a sequence of events that equals sequence  $w$  but for event  $e_i$ , which is changed by  $e_{sup} \in \{e_{sup}\}_{e_i}$ .

At this point, two case exist  $\forall q - l, q(m \in Q_{\mathcal{A}_k})$ :

- $P(q_l, e_{sup}, q_m) = P(q_l, e_i, q_m)$  if  $e_{sup} \in Equ_{q_l, q_m}(e_i)$
- $P(q_l, e_{sup}, q_m) > P(q_l, e_i, q_m)$  if  $e_{sup} \notin Equ_{q_l, q_m}(e_i)$

Thus, for each path  $\theta = (s_0, w'_1, s_1 \dots s_{j-1}, e_i, s_j, \dots w|w|', s_{|w|})$  generating  $w$ , it exists a path  $\theta' = (s_0, w'_1, s_1 \dots s_{j-1}, e_{sup}, s_j, \dots w|w|', s_{|w|})$  generating  $v$  such as:

- if  $e_{sup} \in Equ_{s_{j-1}, s_j}(e_i)$  then

$$\begin{aligned} I(s_0) \times \left( \prod_{h=1}^{|w|} P(s_{h-1}, w'_h, s_h) \right) &= I(s_0) \times \left( \prod_{h=1}^{|w|} P(s_{h-1}, v'_h, s_h) \right) \\ &\rightarrow P(\theta|\mathcal{A}_k) = P(\theta'|\mathcal{A}_k) \end{aligned}$$

- else if  $e_{sup} \notin Equ_{s_{j-1}, s_j}(e_i)$  then

$$\begin{aligned} I(s_0) \times \left( \prod_{h=1}^{|w|} P(s_{h-1}, w'_h, s_h) \right) &< I(s_0) \times \left( \prod_{h=1}^{|w|} P(s_{h-1}, v'_h, s_h) \right) \\ &\rightarrow P(\theta|\mathcal{A}_k) < P(\theta'|\mathcal{A}_k) \end{aligned}$$

Since  $e_{sup} \notin \bigcap_{q_l, q_m \in Q_{\mathcal{A}_k}} Equ_{q_l, q_m}(e_i)$ , the case  $e_{sup} \notin Equ_{s_{j-1}, s_j}(e_i)$  occurs at least one time, thus:

$$P(v|\mathcal{A}_k) = \sum_{\theta' \in \Theta_{\mathcal{A}_k}(v)} P(\theta'|\mathcal{A}_k) > P(w|\mathcal{A}_k) = \sum_{\theta \in \Theta_{\mathcal{A}_k}(w)} P(\theta|\mathcal{A}_k) \quad (\text{A.1})$$

Hence, equation (A.1) proves that, for all sequences  $w$  including an event  $e_i$  with  $\{e_{sup}\}_{e_i} \neq \emptyset$ , it exists another sequence  $v$  having the same length with a greater likelihood. Therefore, event  $e_i$  with  $\{e_{sup}\}_{e_i} \neq \emptyset$  can be excluded for the maximum likelihood computation.

By the same way, we can prove that the likelihood does not change by changing an event  $e_i$  by another event  $e_{eq} \in \{e_{eq}\}_{e_i}$ , then only one of them can be kept in  $\mathcal{A}_k^r$ . Consequently, it holds  $\max_{u \in \Sigma_{\mathcal{A}_k}^{|w|}} [P(u|\mathcal{A}_k)] = \max_{v \in \Sigma_{\mathcal{A}_k}^{|w|}} [P(v|\mathcal{A}_k)]$  and the thesis is proved.  $\square$

**Proposition 4.2** (Chapter 4, Page 89)

Let  $\mathcal{A}_k = \langle Q_{\mathcal{A}_k}, \Sigma_{\mathcal{A}_k}, \delta_{\mathcal{A}_k}, I_{\mathcal{A}_k}, F_{\mathcal{A}_k}, P_{\mathcal{A}_k} \rangle$  be a PFA and  $\mathcal{A}_k^r = \langle Q_{\mathcal{A}_k}, \Sigma_{\mathcal{A}_k}^r, \delta_{\mathcal{A}_k}^r, I_{\mathcal{A}_k}, F_{\mathcal{A}_k}, P_{\mathcal{A}_k}^r \rangle$  be the reduced PFA obtained by the reduction procedure  $\mathcal{A}_k \rightarrow \mathcal{A}_k^r$ . Then the computational complexity of the maximum classical likelihood is the following:

$$C_M = O \left( 2^{[\text{card}(Q_{\mathcal{A}_k})-1]^{|w|}} \times \text{card}(Q_{\mathcal{A}_k})^2 \times |w| \right).$$

*Proof.* We recall that the following properties are direct consequences of equation (4.26):

*Property 1:* If event  $e_i$  is kept using equation (4.26), it exists a set of  $n_1$  origin and destination states  $C_{\mathcal{A}_k}^{n_1} = \{(q_{l_1}, q_{m_1}) \dots (q_{l_p}, q_{m_p}) \dots (q_{l_{n_1}}, q_{m_{n_1}})\}$  such that  $\forall e_j \in \Sigma_{\mathcal{A}_k}, \forall p \in [1, n_1]$  it holds:

$$P[(q_{l_p}, e_i, q_{m_p})] \geq P[(q_{l_p}, e_j, q_{m_p})].$$

Furthermore, according to equation (3.9),  $\tilde{N}(e_i|q_l \rightarrow q_m)$  and  $P(q_l, e_i, q_m)$  do not depend on  $q_l$ . Thus,  $Geq_{q_l, q_m}(e_i)$  and  $Equ_{q_l, q_m}(e_i)$  depend only on  $e_i$  and  $q_m$ . It is possible to rewrite *Property 1* as follow:

---

*Property 2:* If event  $e_i$  is kept using equation (4.26), it exists a set of  $n_2$  destination states  $D_{\mathcal{A}_k}^{n_2} = \{q_{m_1} \dots q_{m_p} \dots q_{m_{n_1}}\}$  such that  $\forall e_j \in \Sigma_{\mathcal{A}_k}, \forall q_l \in Q_{\mathcal{A}_k}, \forall p \in [1, n_2]$  it holds:

$$P[(q_l, e_i, q_{m_p})] \geq P[(q_l, e_j, q_{m_p})].$$

Moreover, for each possible set  $D_{\mathcal{A}_k}^{n_2}$  of destination states, only one event is kept by the equivalent events deletion performed by step 2 of the *reduction procedure*  $\mathcal{A}_k \rightarrow \mathcal{A}_k^r$ . Thus, the number of kept events  $N_{\mathcal{A}_k} = \text{card}(\Sigma_{\mathcal{A}_k}^r)$  is bounded by the number of possible sets  $D_{\mathcal{A}_k}^{n_2}$  that is necessary to evaluate.

For a PFA with  $m = \text{card}(Q_{\mathcal{A}_k})$  states, sets composed with  $n_2 \in [1, m-1]$  destination states can be created. For each  $n_2$ , it exists  $\binom{m-1}{n_2}$  different possible sets  $D_{\mathcal{A}_k}^{n_2}$ . Thus, we have:

$$N_{\mathcal{A}_k} \leq \sum_{i=1}^{m-1} \binom{m-1}{i} = 2^{m-1} - 1.$$

Thus, according to equation (4.22), the complexity of the maximum classical likelihood after the *reduction procedure*  $\mathcal{A}_k \rightarrow \mathcal{A}_k^r$  is the following:

$$C_M = O(\text{card}(\Sigma_{\mathcal{A}_k^r})^{|w|} \times \text{card}(Q_{\mathcal{A}_k})^2 \times |w|).$$

and thus:

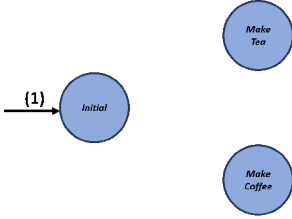
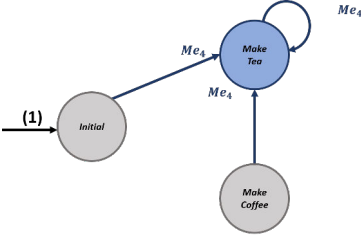
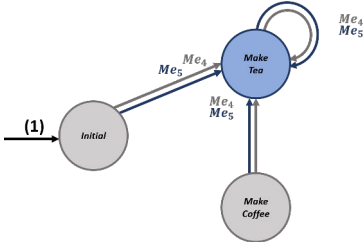
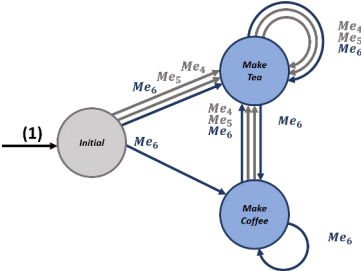
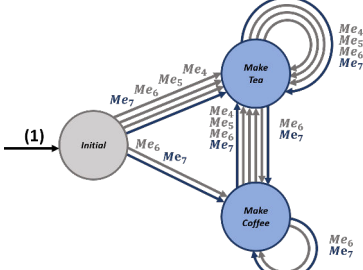
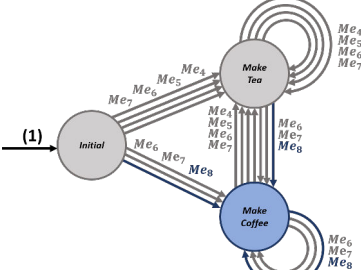
$$C_M = O(\text{card}(2^{[\text{card}(Q_{\mathcal{A}_k)}-1]^{|w|}} \times \text{card}(Q_{\mathcal{A}_k})^2 \times |w|).$$

This proves Proposition 4.2.

□



## Generation of the activities hot beverage preparation and use bathroom PFA structures

<p>States creation <math>Q_{A_2} = \{Make\ tea, Make\ coffee, Initial\}</math></p>	<p>Creation of transitions reflecting the link of the action <i>Make tea</i> with the merge group <math>Me_4</math></p>
	
<p>Creation of transitions reflecting the link of the action <i>Make tea</i> with the merge group <math>Me_5</math></p>	<p>Creation of transitions reflecting the link of the actions <i>Make tea</i> and <i>Make coffee</i> with the merge group <math>Me_6</math></p>
	
<p>Creation of transitions reflecting the link of the actions <i>Make tea</i> and <i>Make coffee</i> with the merge group <math>Me_7</math></p>	<p>Creation of transitions reflecting the link of the action <i>Make Coffee</i> with the merge group <math>Me_8</math></p>
	



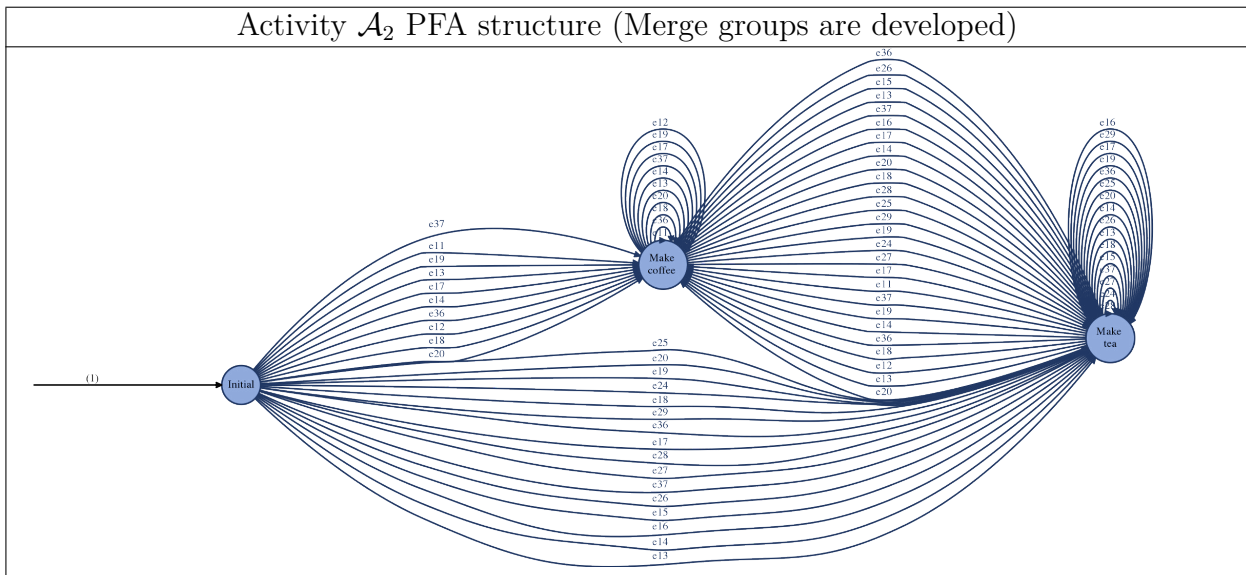


Figure B.0.1: Generation of the activity  $\mathcal{A}_2$  (Hot beverage preparation) PFA structure.

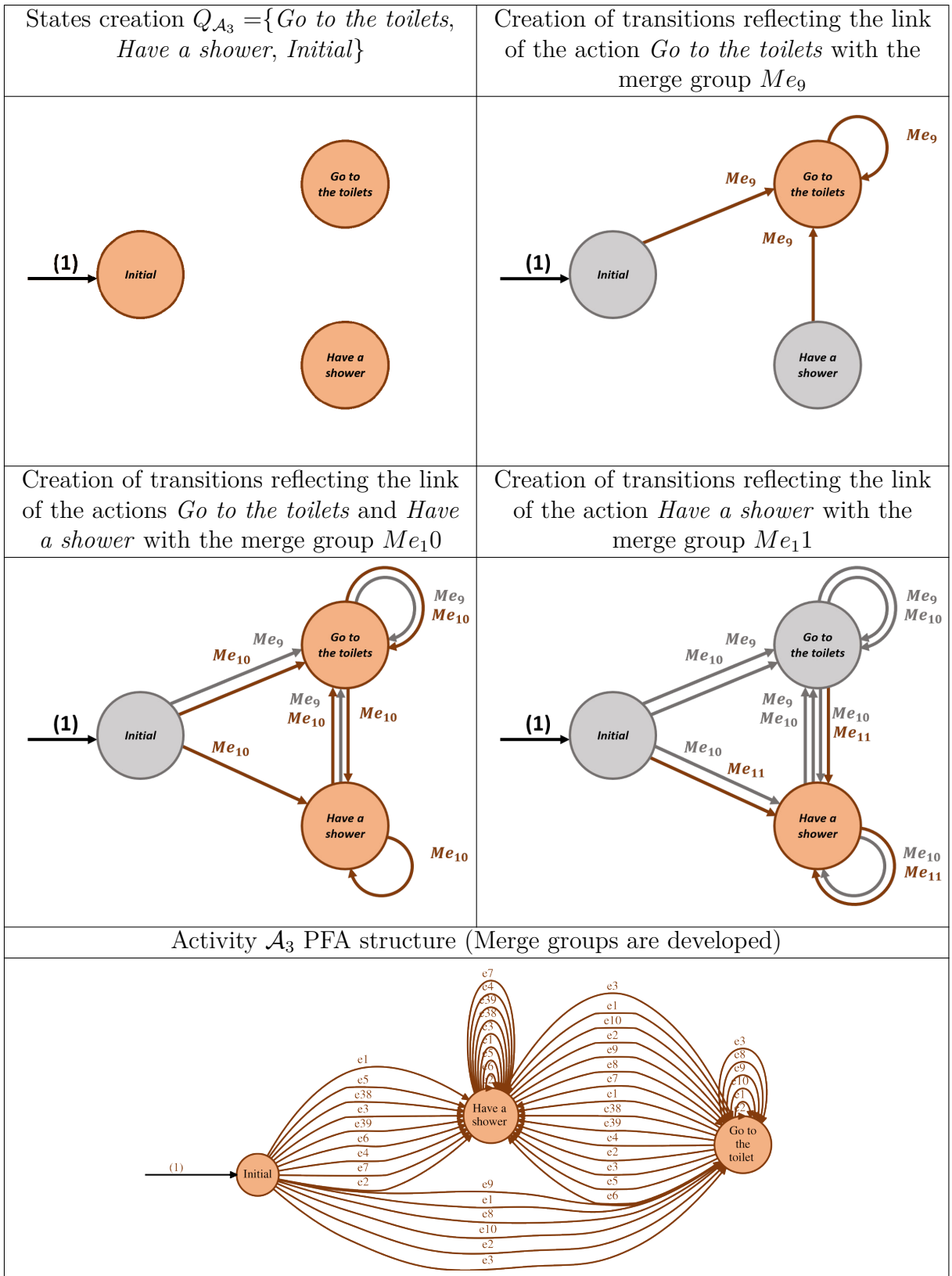


Figure B.0.2: Generation of the activity  $\mathcal{A}_3$  (Use bathroom) PFA structure.



## Counting example during the activities hot beverage preparation and use bathroom performances.

Table C.0.1: Counting example during the activity  $\mathcal{A}_2$  performance.

sequence $w_{\mathcal{A}_2}^{Obs} = e_{36} e_{29} e_{28} e_{23} e_{16} e_{15} e_{18} e_{17}$				
sliding windows	projected sequences	$\mathcal{A}_1$ indicators changes	$\mathcal{A}_2$ indicators changes	$\mathcal{A}_3$ indicators changes
<i>Initial</i>	$\emptyset$	$N_{e_i}^1 = 0$ $\forall e_i \in \Sigma_{\mathcal{A}_1}$	$N_{e_i}^2 = 0$ $\forall e_i \in \Sigma_{\mathcal{A}_2}$	$N_{e_i}^3 = 0$ $\forall e_i \in \Sigma_{\mathcal{A}_3}$
$e_{36}e_{29}e_{28}e_{23}e_{16}$	$w_{p_1}^{Obs} : e_{36}e_{29}e_{28}$ $w_{p_2}^{Obs} : e_{36}e_{29}e_{28}e_{16}$ $w_{p_3}^{Obs} : \emptyset$	$N_{init\ e_{36}}^1 \leftarrow 1$ $N_{e_{36}}^1 \leftarrow 1$ $N_{e_{29}}^1 \leftarrow 1$ $N_{e_{28}}^1 \leftarrow 1$ $N_{e_{36} \rightarrow e_{29}}^1 \leftarrow 1$ $N_{e_{29} \rightarrow e_{28}}^1 \leftarrow 1$	$N_{init\ e_{36}}^2 \leftarrow 1$ $N_{e_{36}}^2 \leftarrow 1$ $N_{e_{29}}^2 \leftarrow 1$ $N_{e_{28}}^2 \leftarrow 1$ $N_{e_{16}}^2 \leftarrow 1$ $N_{e_{36} \rightarrow e_{29}}^1 \leftarrow 1$ $N_{e_{29} \rightarrow e_{28}}^1 \leftarrow 1$ $N_{e_{28} \rightarrow e_{16}}^1 \leftarrow 1$	$\emptyset$
$e_{29}e_{28}e_{23}e_{16}e_{15}$	$w_{p_1}^{Obs} : e_{29}e_{28}$ $w_{p_2}^{Obs} : e_{29}e_{28}e_{16}e_{15}$ $w_{p_3}^{Obs} : \emptyset$	$N_{init\ e_{29}}^1 \leftarrow 1$ $N_{e_{29}}^1 \leftarrow 2$ $N_{e_{28}}^1 \leftarrow 2$ $N_{e_{29} \rightarrow e_{28}}^1 \leftarrow 2$	$N_{init\ e_{29}}^2 \leftarrow 1$ $N_{e_{29}}^2 \leftarrow 2$ $N_{e_{28}}^2 \leftarrow 2$ $N_{e_{16}}^2 \leftarrow 2$ $N_{e_{15}}^2 \leftarrow 1$ $N_{e_{29} \rightarrow e_{28}}^1 \leftarrow 2$ $N_{e_{28} \rightarrow e_{16}}^1 \leftarrow 2$ $N_{e_{16} \rightarrow e_{15}}^1 \leftarrow 1$	$\emptyset$
$e_{28}e_{23}e_{16}e_{15}e_{18}$	$w_{p_1}^{Obs} : e_{28}$ $w_{p_2}^{Obs} : e_{28}e_{16}e_{15}e_{18}$ $w_{p_3}^{Obs} : \emptyset$	$N_{init\ e_{28}}^1 \leftarrow 1$ $N_{e_{28}}^1 \leftarrow 3$	$N_{init\ e_{28}}^2 \leftarrow 1$ $N_{e_{28}}^2 \leftarrow 3$ $N_{e_{16}}^2 \leftarrow 3$ $N_{e_{15}}^2 \leftarrow 2$ $N_{e_{18}}^2 \leftarrow 1$ $N_{e_{28} \rightarrow e_{16}}^1 \leftarrow 3$ $N_{e_{16} \rightarrow e_{15}}^1 \leftarrow 2$ $N_{e_{15} \rightarrow e_{18}}^1 \leftarrow 1$	$\emptyset$

Table C.0.2: Counting example during the activity  $\mathcal{A}_3$  performance.

sequence $w_{\mathcal{A}_3}^{Obs} = e_{39} e_{38} e_2 e_1 e_4 e_7 e_6 e_5$				
sliding windows	projected sequences	$\mathcal{A}_1$ indicators changes	$\mathcal{A}_2$ indicators changes	$\mathcal{A}_3$ indicators changes
<i>Initial</i>	$\emptyset$	$N_{e_i}^1 = 0$ $\forall e_i \in \Sigma_{\mathcal{A}_1}$	$N_{e_i}^2 = 0$ $\forall e_i \in \Sigma_{\mathcal{A}_2}$	$N_{e_i}^3 = 0$ $\forall e_i \in \Sigma_{\mathcal{A}_3}$
$e_{39}e_{38}e_2e_1e_4$	$w_{p_1}^{Obs} : \emptyset$ $w_{p_2}^{Obs} : \emptyset$ $w_{p_3}^{Obs} : e_{39}e_{38}e_2e_1e_4$	$\emptyset$	$\emptyset$	$N_{init}^3 e_{39} \leftarrow 1$ $N_{e_{39}}^3 \leftarrow 1$ $N_{e_{38}}^3 \leftarrow 1$ $N_{e_2}^3 \leftarrow 1$ $N_{e_1}^3 \leftarrow 1$ $N_{e_4}^3 \leftarrow 1$ $N_{e_{39} \rightarrow e_{38}}^3 \leftarrow 1$ $N_{e_{38} \rightarrow e_2}^3 \leftarrow 1$ $N_{e_2 \rightarrow e_1}^3 \leftarrow 1$ $N_{e_1 \rightarrow e_4}^3 \leftarrow 1$
$e_{38}e_2e_1e_4e_7$	$w_{p_1}^{Obs} : \emptyset$ $w_{p_2}^{Obs} : \emptyset$ $w_{p_3}^{Obs} : e_{38}e_2e_1e_4e_7$	$\emptyset$	$\emptyset$	$N_{init}^3 e_{38} \leftarrow 1$ $N_{e_{38}}^3 \leftarrow 2$ $N_{e_2}^3 \leftarrow 2$ $N_{e_1}^3 \leftarrow 2$ $N_{e_4}^3 \leftarrow 2$ $N_{e_7}^3 \leftarrow 1$ $N_{e_{38} \rightarrow e_2}^3 \leftarrow 2$ $N_{e_2 \rightarrow e_1}^3 \leftarrow 2$ $N_{e_1 \rightarrow e_4}^3 \leftarrow 2$ $N_{e_4 \rightarrow e_7}^3 \leftarrow 1$
$e_2e_1e_4e_7e_6$	$w_{p_1}^{Obs} : \emptyset$ $w_{p_2}^{Obs} : \emptyset$ $w_{p_3}^{Obs} : e_2e_1e_4e_7e_6$	$\emptyset$	$\emptyset$	$N_{init}^3 e_2 \leftarrow 1$ $N_{e_2}^3 \leftarrow 3$ $N_{e_1}^3 \leftarrow 3$ $N_{e_4}^3 \leftarrow 3$ $N_{e_7}^3 \leftarrow 2$ $N_{e_6}^3 \leftarrow 1$ $N_{e_2 \rightarrow e_1}^3 \leftarrow 3$ $N_{e_1 \rightarrow e_4}^3 \leftarrow 3$ $N_{e_4 \rightarrow e_7}^3 \leftarrow 2$ $N_{e_7 \rightarrow e_6}^3 \leftarrow 1$
$e_1e_4e_7e_6e_5$	$w_{p_1}^{Obs} : \emptyset$ $w_{p_2}^{Obs} : \emptyset$ $w_{p_3}^{Obs} : e_1e_4e_7e_6e_5$	$\emptyset$	$\emptyset$	$N_{init}^3 e_1 \leftarrow 1$ $N_{e_1}^3 \leftarrow 4$ $N_{e_4}^3 \leftarrow 4$ $N_{e_7}^3 \leftarrow 3$ $N_{e_6}^3 \leftarrow 2$ $N_{e_5}^3 \leftarrow 1$ $N_{e_1 \rightarrow e_4}^3 \leftarrow 4$ $N_{e_4 \rightarrow e_7}^3 \leftarrow 3$ $N_{e_7 \rightarrow e_6}^3 \leftarrow 2$ $N_{e_6 \rightarrow e_5}^3 \leftarrow 1$

*"Ah, so tired. I'll rest for just a moment."*  
- XT-002 Deconstructor -  
World of Warcraft, *The siege of Ulduar*



# Modélisation et reconnaissance des habitudes de vie d'un habitant dans un habitat intelligent

## Résumé

La plupart des travaux réalisés dans le domaine de l'assistance à l'autonomie à domicile (AAL) reposent sur l'utilisation de capteurs visuels et audio tels que des caméras. Or, ces capteurs sont souvent rejetés par le patient à cause de leur caractère invasif. Des approches alternatives requièrent l'utilisation de capteurs embarqués sur la personne (GPS, bracelets électroniques ou puces RFID dans les vêtements...), et leur pertinence est donc ramenée à l'hypothèse que les personnes les portes effectivement, sans jamais les rejeter ni les oublier.

Pour ces raisons que, dans cette thèse, nous trouvons plus pertinentes les approches uniquement basées sur l'utilisation de capteurs binaires intégrés dans l'habitat, tels que les détecteurs de mouvements, les tapis sensitifs ou les barrières optiques.

Dans un tel contexte technologique, il devient intéressant à utiliser les paradigmes, les modèles et les outils des systèmes à événement discrets (SED), initialement plutôt développés pour la modélisation, l'analyse et la commande des systèmes industriels complexes.

Dans ces travaux de thèse, l'objectif est de construire une approche pour la modélisation et le suivi des habitudes de vie, basée sur les modèles et le paradigme des SED et répondant à une problématique qui s'énonce de la manière suivante:

*L'objectif est de développer un cadre global pour découvrir et reconnaître les activités de la vie quotidienne d'un habitant vivant seul dans une maison intelligente. Cette maison intelligente doit être équipée uniquement de capteurs binaires, l'étiquetage par des experts des activités observées ne doit pas être nécessaire et les activités peuvent être représentées par des modèles probabilistes.*

La première méthode présentée dans cette thèse permet de construire un modèle d'automate à états finis probabiliste (PFA) à partir d'une base de données d'apprentissage et d'une description experte des activités à modéliser listées par des médecins.

La seconde méthode développée lors de cette thèse estime, au vu des observations, quelle activité la personne observée réalise.

Les méthodes décrites dans cette thèse sont illustrées en utilisant les données générées localement via l'utilisation d'un appartement prêté par ENS Paris-Saclay équipé pour répondre aux besoins expérimentaux de cette thèse.





# Modelling and Recognition of Human Activities of Daily Living in a Smart Home

## Abstract

Most of the work done in the field of ambient assisted living (AAL) is based on the use of visual and audio sensors such as cameras. However, these sensors are often rejected by the patient because of their invasiveness. Alternative approaches require the use of sensors embedded in the person (GPS, electronic wristbands or RFID chips in clothing ...), and their relevance is therefore reduced to the assumption that people actually wear them, without rejecting nor forgetting them.

For these reasons, in this thesis, we find more relevant the approaches based on the use of binary sensors integrated into the habitat only, such as motion detectors, sensory mats or optical barriers.

In such a technological context, it becomes interesting to use paradigms, models and tools of Discrete Event Systems (DES), initially developed for modelling, analysis and control of complex industrial systems.

In this thesis work, the goal is to build an activity of daily living modelling and monitoring approach, based on the models and the paradigm of the DES and answering a problem that is expressed as follows:

*The objective is to develop a global framework to discover and recognise activities of daily living of an inhabitant living alone in a smart home. This smart home has to be equipped with binary sensors only, expert labelling of activities should not be needed and activities can be represented by probabilistic models.*

The first method presented in this thesis allows to build a probabilistic finite-state automata (PFA) from a learning database and an expert description of the activities to be modelled given by the medical staff.

The second method developed during this thesis estimates, according to the observations, the activity performed by the monitored inhabitant.

The methods described in this thesis are applied on data generated using an apartment lent by ENS Paris-Saclay and equipped according to the experimental needs of this thesis.



# Modellizzazione e riconoscimento delle abitudini di un abitante in una casa intelligente

## Sintesi

La maggior parte dei lavori nel settore dell’Ambient Assisted Living (AAL) si basa sull’uso di sensori visivi e audio come le telecamere. Tuttavia, questi sensori sono spesso rifiutati dal paziente a causa della loro natura invasiva. Gli approcci alternativi richiedono l’uso di sensori integrati nella persona (GPS, bracciali elettronici o chip RFID...), e la loro rilevanza è quindi ridotta all’ipotesi che le persone li indossino effettivamente, senza mai rifiutarli o dimenticarli.

Per questi motivi, in questa tesi, troviamo approcci più rilevanti basati esclusivamente sull’uso di sensori binari integrati nell’habitat, come rilevatori di movimento, tappeti sensoriali o barriere fotoelettriche.

In tale contesto tecnologico, diventa interessante utilizzare i paradigmi, i modelli e gli strumenti dei sistemi ad eventi discreti (SED), inizialmente sviluppati per la modellazione, l’analisi e il controllo di sistemi industriali complessi.

In questo lavoro di tesi, l’obiettivo è quello di presentare un metodo per la modellazione e il monitoraggio delle abitudini di vita, basato sui modelli e paradigmi di SED e rispondendo ad un problema che si esprime come segue:

*L’obiettivo è quello di sviluppare un quadro globale per rivelare e riconoscere le attività della vita quotidiana di una persona che abita da sola in una smart home che dovrebbe essere dotata solo di sensori binari. Inoltre si suppone che non sia necessaria l’etichettatura delle attività osservate da parte di un esperto e tali attività sono rappresentate da modelli probabilistici.*

Il primo metodo presentato in questa tesi permette di costruire un modello probabilistico di automa a stati finiti (PFA) ottenuto da un database di apprendimento e una descrizione delle attività da parte di medici.

Il secondo metodo sviluppato in questa tesi stima, alla luce delle osservazioni, quale attività svolge la persona osservata.

I metodi descritti sono illustrati utilizzando dati generati localmente attraverso l’uso di un appartamento messo a disposizione da ENS Paris-Saclay e attrezzato per soddisfare le esigenze sperimentali di questa tesi.

