

# Modelling Activities of Daily Living with Petri nets

Matias Garcia-Constantino

School of Computing  
Ulster University

Jordanstown, United Kingdom  
m.garcia-constantino@ulster.ac.uk

Alexandros Konios

School of Computing, Electronics and Mathematics  
Coventry University

Coventry, United Kingdom  
alexandros.konios@coventry.ac.uk

Chris Nugent

School of Computing  
Ulster University

Jordanstown, United Kingdom  
cd.nugent@ulster.ac.uk

**Abstract**—Modelling Activities of Daily Living (ADLs) is an important step in the process to design and implement reliable sensor systems that effectively monitor the activities of the ageing population. Once modelled, unusual activities may be detected that have the potential of impacting upon a person's well-being. The use of Petri nets to model ADLs is considered in this research as a means to capture the intricate behaviours of ambient systems. To our best knowledge there has not been extensive work in the related literature, hence the novelty of this work. The ADLs considered in the developed Petri net model are: (i) preparing tea, (ii) preparing coffee, and (iii) preparing pasta. The first two ADLs listed are deemed to have many occurrences during a typical day of an elderly person. The third activity is representative of activities that involve cooking. Hence, abnormal behaviour detected in the context of these activities can be an indicator of a progressive health problem or the occurrence of a hazardous incident. The completion and non-completion of activities are considered in the developed Petri net model and are also formally verified. The description of the sensor system of the kitchen ADLs, its Petri net model and verification results are presented. Results show that the Petri net modelling of ADLs can reliably and effectively reflect the real behaviour of the examined system detecting all the activities of the users that can exhibit both their normal and abnormal behaviour.

**Index Terms**—Activities of Daily Living, Petri nets, Modelling, Verification

## I. INTRODUCTION

Modelling Activities of Daily Living (ADLs) is important to design and implement reliable sensor systems that effectively monitor the activities of the ageing population and have the ability to detect unusual activities that may affect a person's well-being. The reasons for considering kitchen ADLs of elderly people are: (i) they are typically completed many times during a day, and (ii) the detection of abnormal behaviour can be an indicator of a progressive health problem taking place (dementia, osteoporosis, arthritis, etc.) or the occurrence of a hazardous incident (falls, burns, cuts, food or smoke intoxication, etc.). One of the advantages of monitoring ADLs is the unobtrusiveness of the sensors used, which can be placed within the environment and in appliances of interest. This avoids the requirement for the use of wearable technologies.

One way to identify normal or abnormal behaviour is to evaluate the completion or non-completion of an activity respectively. In the context of ADLs, an activity is regarded as being successfully complete if the desired output has been obtained following the typical steps defined for that activity. This can be illustrated with the example of a person preparing a cup of tea performing the following steps: going to the kitchen, getting a cup, getting and putting a tea bag inside the cup, pouring hot water in the cup and finally, pouring milk into the cup. An activity is regarded as incomplete if the user begins following the steps for the activity, however, at some point there is a diversion towards other steps that correspond to another activity.

**ACKNOWLEDGEMENT:** Invest Northern Ireland is acknowledged for supporting this project under the Competence Centre Programs Grant RD0513853 - Connected Health Innovation Centre.

This can be the case, for example, of a person following the steps of preparing a cup of tea, however, not completing the activity.

If incomplete activities are detected by sensors on a regular basis, then it could be inferred that the person has a health problem and must be attended by carers. In addition, the sensor system could send prompts to the user suggesting the completion of an activity. In the case of active appliances which could represent a hazard when unattended (e.g. stove or oven), alert messages could be sent to the user, relatives and/or carers.

The use of Petri nets to model ADLs, which is the main contribution of this paper, is considered as a means to capture the intricate behaviours of ambient systems, such as the activities described above. The granularity that Petri nets models offer allows for a complete and detailed understanding of the different variations and cases of the ADLs modelled.

The remainder of the paper is organised as follows: Section II presents the related work in the areas of activity recognition and formal analysis of intelligent systems. Section III describes the ADLs considered for Petri net modelling. Section IV describes the sensors used in the system. Section V outlines the Petri net model for the kitchen ADLs of interest and Section VI discusses the verification results. Finally, Section VII presents the conclusions.

## II. BACKGROUND

As discussed in [4], one of the ways in which activity recognition can be classified is in terms of vision-based versus sensor-based. While the former can provide more information to an activity recognition system, it has the disadvantage of not protecting the privacy of the users. Activity recognition using sensors is typically classified in terms of wearable sensors versus dense sensors. Wearable sensors can be worn by users in parts of their body or in their clothes [6], [9], [21]. While wearable sensors have the advantages of monitoring and collecting data regardless of the location of the users, some of the main disadvantages are that the users are responsible for their correct use and for charging the battery. Dense sensors are attached to objects in the environment with which the user interacts (e.g. kitchen, bathroom, kettle) [20]. Dense sensors have the advantages of not being intrusive and typically do not require the user to charge them periodically. However, the main disadvantages of dense sensors are that they need to be well placed in the environment and within an adequate setting.

Regarding activity modelling, [4] mentions that it can be classified in terms of data-driven versus knowledge-driven. The former can be categorised as generative (using a complete description of the input) or discriminative (mapping inputs to outputs). Knowledge-driven activity modelling considers that the objects used in ADLs are limited even if the sequence of actions to perform an activity varies. Three main approaches are identified in [4]: (i) mining-based (using information retrieval), (ii) logic-based (using formal

knowledge modelling), and (iii) ontology-based (using learning from datasets).

The activity recognition is a challenging process that is related to the operation of the intelligent systems. Once the activities have been identified, then they can be considered in the structural and behavioural analysis of the system to collect useful observations with respect to the users' and system's actions. Traditionally, one of the ways to examine and analyse real-world systems is via the use formal methods [22]. Formal methods are well-defined and rigorous mathematical techniques that consider different formalisms and semantics for the modelling and verification of systems that exhibit either sequential or concurrent behaviour [2], [17], [19]. In this paper, we use Petri nets for the analysis of the smart kitchen activities as Petri nets modelling falls within the logic-based category mentioned earlier. Moreover, to our best knowledge there has not been extensive work in the literature on activity modelling using Petri nets in the context of ADLs.

Petri nets are a mathematical modelling language that support the graphical representation of system behaviour. Since their introduction by C.A. Petri in 1962 [17], Petri nets have been successfully used for the modelling and analysis of several interactive and concurrent systems facilitating these demanding processes. Example of systems that have been examined with the help of Petri nets are also systems that incorporate ambient intelligence [11], [15], [16].

For the modelling of the kitchen ADLs, we use a subclass of Colour Petri Nets (CPNs) [10], the Ambient Petri Nets (APNs) [12], [13], which can effectively capture the interactive behaviour of the smart kitchen and the functioning of its sensors. The use of the APNs class of Petri nets facilitates the modelling of the smart kitchen through the compositionality of a fundamental modular APN net (basic step net) [12], which represents a single interaction between the user and the environment. Specifically, APNs through the compositionality and modularity of the basic step nets represent the behaviour of the smart kitchen as a 'collection' of interactive individual actions that are performed by the users and are followed by the non-intrusive response of the environment.

Furthermore, the use of the APN class of Petri nets for the modelling process enhances the analysis of the intelligent systems by combining the qualitative reasoning of Petri nets and the analysis of system properties through the use of model checking [14].

To check the correctness of the model with respect to the system specification, we examine fundamental properties of the system, which are also related to both the architecture and behaviour of the system [7]. Specifically, the verification of the developed model for the smart kitchen considers the examination of safety, liveness and general functioning properties that are associated with the basic components, features and operations of the system, such as sensors' operation and completion of ADLs. All these properties are expressed in Computation Tree Logic (CTL) [5] and are examined using the Charlie model checker [1].

### III. KITCHEN ADLs CONSIDERED FOR PETRI NET MODELLING

The ADLs considered in the developed Petri net model are: (i) preparing tea, (ii) preparing coffee, and (iii) preparing pasta. These kitchen ADLs are based on the ones used in [3], where they were modelled using ontologies. These kitchen ADLs fall into two categories: (i) preparing a hot drink, and (ii) preparing a hot meal. It is acknowledged that there can be a variety in number and order of steps to perform the aforementioned ADLs. We have considered all the main steps involved to perform these activities and different

orders in which they can be followed. In our model, the steps and order required for each of the ADLs are as follows:

- Preparing tea
  - GoToKitchen, GetCup, GetTea, PourHotWater, GetMilk, GetSugar
  - GoToKitchen, GetCup, PourHotWater, GetMilk, GetTea, GetSugar
  - GoToKitchen, GetCup, GetMilk, GetTea, PourHotWater, GetSugar
- Preparing coffee
  - GoToKitchen, GetCup, GetCoffee, PourHotWater, GetMilk, GetSugar
  - GoToKitchen, GetCup, PourHotWater, GetMilk, GetCoffee, GetSugar
  - GoToKitchen, GetCup, GetMilk, GetCoffee, PourHotWater, GetSugar
- Preparing pasta
  - GoToKitchen, GetPan, PourWater, GetSalt, BoilWater, GetPasta, CookPasta, RemoveWaterFromPan, GetSauce
  - GoToKitchen, GetPan, PourWater, BoilWater, GetPasta, CookPasta, RemoveWaterFromPan, GetSauce

For the purpose of the Petri net modelling, it is assumed that the number of steps required for each activity is constant and that the existing steps cannot be removed and more steps cannot be added. The two main differences between the "preparing tea" and "preparing coffee" activities are: (i) the tea bag is usually removed from the cup, and (ii) while adding milk to tea is required, adding milk to coffee is optional. These differences are explicitly captured by the produced Petri net model.

### IV. SENSOR SYSTEM FOR KITCHEN ADLs

The sensor system to detect kitchen ADLs that was considered and modelled with Petri nets in this paper is the one at the smart kitchen in the Smart Environments Research Group (SERG)<sup>1</sup> at Ulster University (see layout in Figure 1).

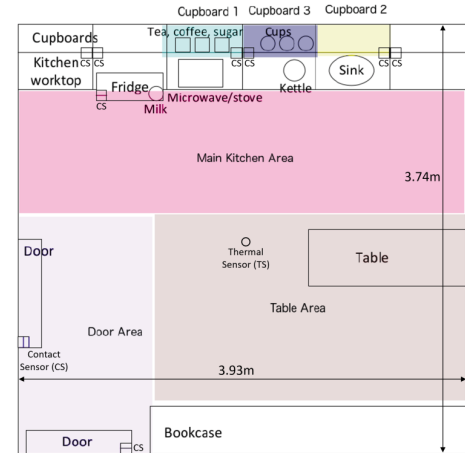


Fig. 1: Smart kitchen layout at Smart Environments Research Group.

In this paper, two types of sensors are considered for monitoring the kitchen ADLs: (i) contact sensor, and (ii) thermal sensor. A number of contact sensors are used in conjunction with a thermal sensor to support the identification of ADLs. The contact sensors were attached

<sup>1</sup><https://www.ulster.ac.uk/research/institutes/computer-science/groups/smart-environments/about>

to objects with which the user has interaction in the kitchen in the context of the ADLs considered in this paper: doors, cups, kettle, pan, refrigerator, cupboards and containers (tea, coffee, sugar, milk, salt and pasta). The contact sensors combine wireless transmitters and magnetic switches. The signals from the contact sensors have two possible states (on or off) and are monitored and collected by SensorCentral [18], a sensor data platform, for further processing of data analysis. The contact sensors (CS) are represented in Figure 1 as rectangles divided into two parts that can be separated ('on' state) or joint ('off' state).

In this model, it is assumed that a thermal sensor (TS) is mounted in the ceiling of the kitchen in a central position where it can identify three main kitchen areas where the users can be detected: (i) main kitchen area, (ii) table area, and (iii) areas around the doors (refer to Figure 1). The thermal sensor has a resolution of 32x31, a 90° by 86° field of view that provides a coverage area of 6m by 6m at a height of 2.5m, and a sample rate of 10Hz. Note that the images captured by the thermal sensor do not have enough resolution to be considered as privacy invasive. Figure 2(a) shows the thermal sensor and Figure 2(b) shows a thermal image of a person sitting. The data from the thermal sensor is also monitored and collected in SensorCentral. The main purposes of using a thermal sensor in this sensor system are: (i) to identify (in an object classification sense) participants that interact with objects, (ii) to identify the positions of the participants with respect to the kitchen area, and (iii) to identify objects that emit heat when they are being used, such as kettles or pans.

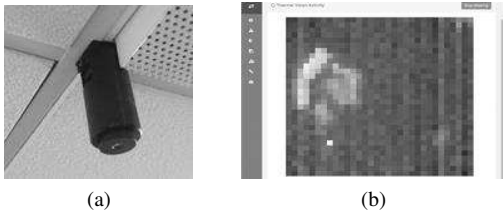


Fig. 2: (a) Thermal sensor. (b) Example of thermal sensor image.

Thus, the basic idea is to use the contact sensors and the thermal sensor to detect the interactions of the user with the objects and when some of these objects are used. The objects being interacted with and the sequence of the interactions support the identification of ADLs by identifying the participants that interact with the objects and their location with respect to the kitchen area. When preparing a drink or meal, then the user is considered to be moving around the main kitchen area (in front of cupboards and main appliances). If the user has finished with the preparation of the drink or meal, then it is assumed that the user uses the table to drink the beverage (tea or coffee) or to eat the meal (pasta) respectively. Finally, a user is in the doors' area when is entering or leaving the kitchen.

The sequence of main user-object interactions identified by the contact sensors for the ADLs are all considered in the Petri net representation. The model produced explicitly exhibits the following behaviour, interactions and sub-interactions:

- 1) GoToKitchen: *OpenKitchenDoor, CloseKitchenDoor*
- 2) GetCup: *OpenCupboardDoor, TakeCupFromCupboard, CloseCupboardDoor*
- 3) GetPan: *OpenCupboardDoor, TakePanFromCupboard, CloseCupboardDoor*
- 4) GetCoffee: *OpenCupboardDoor, TakeCoffeeFromCupboard, CloseCupboardDoor, (PutCoffeeInCup)*
- 5) GetTea: *OpenCupboardDoor, TakeTeaFromCupboard, CloseCupboardDoor, (PutTeaBagInCup)*

- 6) GetSugar: *OpenCupboardDoor, TakeSugarFromCupboard, CloseCupboardDoor, (PutSugarInCup)*
- 7) GetMilk: *OpenFridgeDoor, TakeMilkFromFridge, CloseFridgeDoor, AddMilkToCup*
- 8) GetPasta: *OpenCupboardDoor, TakePastaFromCupboard, CloseCupboardDoor, (AddPastaToPan)*
- 9) GetSauce: *OpenCupboardDoor, TakeSauceFromCupboard, CloseCupboardDoor, (AddSauceToPan)*
- 10) PourHotWater: *FillKettleWithWater, StartKettle, PourHotWater*
- 11) BoilWater: *TurnOnStove, HeatWaterUntilBoiling*
- 12) CookPasta: *KeepPastaInPanUntilCooked, TurnOffStove*
- 13) RemoveWaterFromPan: *TiltPanToPourWater, ReturnPanToFlatSurface*

The activities that are not explicitly identified by the contact sensors (those in parenthesis) are also modelled, assuming that these actions will eventually occur; contributing in that way towards the completion of the ADLs. Furthermore, all the above interactions and sub-interactions are combined in the developed models in order to show the sequence of actions required to complete the ADLs. Finally, all the objects involved in the user-object interactions described above and which have contact sensors attached to them are included in the behavioural model of the kitchen ADLs, these are: (i) kitchen door, (ii) cupboard door, (iii) cup, (iv) kettle, (v) pan, (vi) coffee container, (vii) tea container, (viii) sugar container, (ix) salt container, (x) pasta container, (xi) sauce container, (xii) milk container, and (xiii) fridge. The containers can be in the form of bags, jars or bottles.

## V. PETRI NET MODEL FOR KITCHEN ADLS

The developed Petri net model involves two users for the sake of simplicity, however, is fully scalable and can be easily extended by adding more tokens to the appropriate places of the model representing the additional users and respective resources needed for the executed activities. Furthermore, since the model is considered as a collection of modular actions that are composed to capture the entire behaviour of the smart kitchen with respect to the ADLs, every basic step net actually depicts how an action that takes place is detected by the respective system sensors (either contact or thermal) or how it contributes towards the completion of each task (refer to Figure 3).

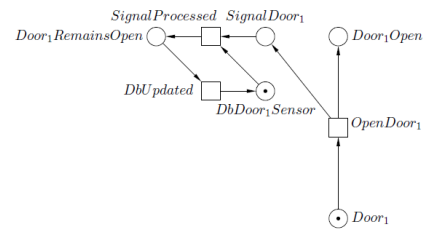


Fig. 3: Example of a basic step net representing the action of opening the kitchen door.

The model also considers that the users interacting with the smart kitchen have always the initiative and can freely decide whether they will complete an ongoing activity or not. This gives a more realistic aspect to the developed model with reference to the functioning of the examined system (e.g. a user is never forced by the kitchen to perform action). For instance, a user can initially go to the kitchen wanting to make a drink. He/She starts its preparation and then he/she changes his/her mind starting a meal preparation and abandoning the idea of a drink. This means that the user is free to do whatever he/she wants when interacting with the system. For modelling purposes, the only restriction that has been imposed on the users' actions is that they

can only leave the kitchen once they have completed at least one of the main activities. This requirement has been set to show that some activities can be replaced by some others or can be completed in conjunction with others. Finally, incorporating the initiative of the users into our model enables us to detect or define patterns of actions that indicate an abnormal behaviour that could be associated with a potential health issue or hazardous event.

To represent the ADLs of the smart kitchen, the following main case scenarios have been considered during the modelling process:

- 1) A user successfully makes a tea, then he/she drinks his/her tea sitting around the table and finally leaves the kitchen.
- 2) A user successfully makes a coffee, then he/she drinks his/her coffee sitting around the table and finally leaves the kitchen.
- 3) A user successfully makes a tea and a meal, then he/she drinks and eats sitting around the table and finally leaves the kitchen.
- 4) A user successfully makes a coffee and a meal, then drinks and eats sitting around the table and finally leaves the kitchen.
- 5) A user successfully makes a meal, then he/she eats his/her pasta sitting around the table and finally leaves the kitchen.
- 6) A user successfully makes a meal, then proceeds with tea preparation but he/she never completes that. Then, he/she eats pasta sitting around the table and finally leaves the kitchen.
- 7) A user successfully makes a meal, then he/she proceeds with coffee preparation but never completes that. Then, he/she eats pasta sitting around the table and finally leaves the kitchen.
- 8) A user gets the ingredients for both tea and coffee but he/she can make only one of them. Then, he/she drinks it at the table and finally exits the kitchen.

It is worth mentioning that every single scenario above consists of a sequence of actions that can be performed in a different order to complete those activities (as is specified in Section III) and is also based on the interactions and sub-interactions detected by the contact and thermal sensors of the system (see Section IV). Now, regarding the execution of the actions required to complete an ADL activity, the model represents both the sequential and concurrent behaviour of the smart kitchen. This means that a user can only perform sequential actions with respect to his/her activities, but he/she can execute actions concurrently to other users' activities.

The concurrent behaviour of the system can be represented by using the eight scenarios mentioned earlier, but considering that both users follow them. Another way to examine the concurrent behaviour of the system through the model is by using a combination of the aforementioned scenarios. This means that they can perform same or different activities at the same time (i.e. concurrently to each other). Additionally to the previous scenarios, you could take into consideration the following scenario, which clearly demonstrates the concurrent behaviour of the system modelled:

- 9) One user successfully makes a coffee and the other successfully prepares a meal, then both users sit around the table drinking and eating the coffee and the meal respectively. Finally both leave the kitchen.

Finally, to exhibit the behaviour described in the scenarios, the following assumptions have been reflected on the model:

- 1) Once a user enters the kitchen, he/she can move around the main kitchen area, the table area and the door area, as shown in Figure 1. The user is supposed to move from one area to another with the order specified below:
  - (i) Door Area (Entering) → Main Kitchen Area → Table Area → Door Area (Exiting)

This assumption is made to model the detection of the users from the thermal sensor located on the ceiling of the kitchen.

- 2) Doors are left open while the users are in the kitchen. A door closes only when the user exits the kitchen.
- 3) A user enters and exits the kitchen using the same door. This assumption has been made to examine whether the sensors of both doors can potentially be in 'on' and 'off' states representing the open and close condition respectively.
- 4) Once the preparation of the meal has started, the user is required to finish it. Moreover, while he/she is cooking, that specific user cannot make coffee or tea. This assumption derives from the fact that it is less likely for a user to pause or leave an activity incomplete like cooking in comparison to the other two activities.
- 5) Once the preparation of coffee or tea has started, the user can pause, stop or abandon the process by starting the preparation of a meal. Afterwards, that user can make the drink if he/she wants or can just leave the kitchen having eaten the meal first.
- 6) A user can make only one drink each time that he/she uses the kitchen. He/She can either make coffee or tea, however, not both. This assumption was made on the basis that the user usually drinks only one drink and also on the basis that he/she can make a drink only for himself/ herself.
- 7) The quantity of the ingredients used for making coffee or tea is limited. For example, each user can only have one coffee or tea per system run. Similarly, for the meal preparation the number of pans is one for each user. In this model the number of pans is two, thus, each user can make only one pasta meal.
- 8) It is assumed that only the quantity of ingredients needed to make a drink (tea or coffee) are taken out of the respective cupboards. The containers of these ingredients are put back to them. It is also assumed that some of these ingredients will eventually be used in the kitchen ADLs.
- 9) A user can take more ingredients than he/she needs for the drink preparation. The ingredients that will not be used are supposed to be left out of the cupboards without being put back into it.
- 10) The cupboards (see Figure 1) contain: (i) Cupboard 1: Tea, Coffee and Sugar Containers, (ii) Cupboard 2: Pan, pasta container, salt and sauce, and (iii) Cupboard 3: Cups.
- 11) Fridge contains only the milk container/bottle. As with the other ingredients, the user gets only the quantity needed for making the coffee or tea.

All the above information is taken into account for the development of the smart kitchen model that captures the three main ADL activities. As already been mentioned, the main objective of the produced model is to detect potential abnormalities with respect to the execution of the kitchen ADLs. It should be noted that the development of the model has been conducted using the Snoopy modelling tool [8]. A small example of how a coffee activity can be modelled using Snoopy modelling tool is illustrated in Figure 4.

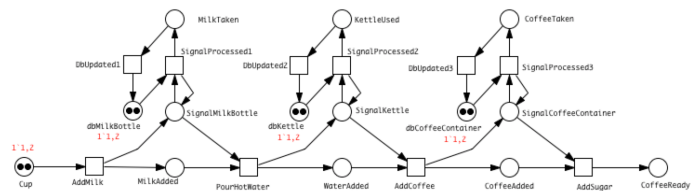


Fig. 4: Example of modelling a coffee activity using Snoopy.

Figure 4 shows the sequence of actions required in order to make

a coffee<sup>2</sup>. This chunk of the system behavioural model can depict both the sequential and concurrent behaviour of the system as the two tokens correspond to the two users of the smart kitchen who can execute the same activity either sequentially (i.e. one after the other) or concurrently (i.e. at the same time). Finally, it is worth mentioning that the entire model of the smart kitchen consists of all the different sequences of actions for the performance of the ADLs. This implies that the model captures both the sequential and concurrent behaviour of the environment.

## VI. PETRI NET MODEL VERIFICATION RESULTS

In this Section, we investigate whether the produced model exhibits the behaviour required by its specification in order to enable the detection of normal and abnormal behaviour of users with respect to the completion of the activities or not. To demonstrate this, we check system properties that are related to the core scenarios, as described in Section V. These properties are expressed in CTL by using its standard *path quantifiers* (i.e. **A** and **E**) and *temporal operators* (i.e. **G**, **F**, **X** and **U**). Joining them together we get combinations like **AG**, **AF**, **AX**, **EG**, **EF** and **EX**, which can be used for the formation and examination of the system properties [5]. Further to the examination of the core scenarios, we also examine sub-scenarios that consider cases where the user performs actions that do not contribute towards the completion of the activities. An example of such cases is that the user opens and closes the cupboards without taking anything from inside.

First, we verify all the properties of the system that are linked to the successful completion of the three ADLs considered in the model, i.e. making coffee, making tea and preparing a meal. These properties usually indicate a normal behaviour of the users when they interact with the environment. Note that the places *CoffeeReady*, *TeaReady* and *PastaReady* used in the following propositions correspond to ‘states’ of the APN model that can be visited only after the sequence of actions required for the completion of these activities has been followed successfully. For further comprehension refer to Figure 4.

Starting with the examination of the case expressed by the first clause of the list of scenarios provided, we provide the CTL proposition that is used for the model checking.

$$\phi_1 = EF(Cup \rightarrow TeaReady \rightarrow EX(AtTableDrinking \rightarrow Exiting))$$

Checking  $\phi_1$  with Charlie model checker, it results that the model exhibits the desired behaviour as the output is true, implying that the property is satisfied. If a user executes this activity following a sequence of actions, the system considers it as a normal behaviour.

The second scenario refers to the successful preparation of coffee and is expressed as a CTL proposition below:

$$\phi_2 = EF(Cup \rightarrow CoffeeReady \rightarrow EX(AtTableDrinking \rightarrow Exiting))$$

The property  $\phi_2$  is similar to  $\phi_1$  and its model checking result (i.e. true) proves that this property is also satisfied denoting another activity where the user exhibits a normal behaviour.

The next property examines the case where a user prepares both tea and meal. This is expressed in CTL as follows:

$$\phi_3 = EF(TeaReady \wedge PastaReady \rightarrow EX(FinishDrinking \wedge FinishingEating \rightarrow Exiting))$$

<sup>2</sup>The model provided in Figure 4 is a tiny part of the model representing the first sequence of actions for preparing coffee as described in Section III.

Figure 5 presents the output of the model checker for this property, which indicates that a combination of activities like preparing tea and meal can be successfully conducted by a user.

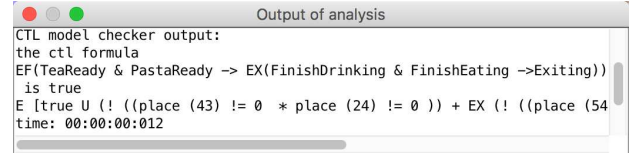


Fig. 5: Model checking output for the  $\phi_3$  property.

Similarly for the case where a user can successfully prepare both a coffee and a meal (refer to CTL expression below).

$$\phi_4 = EF(CoffeeReady \wedge PastaReady \rightarrow EX(FinishDrinking \wedge FinishingEating \rightarrow Exiting))$$

In both cases, a user interacts with the smart kitchen completing these activities without revealing any abnormal behaviour.

Next, property  $\phi_5$  examines the completion of the meal preparation activity. In this case, this activity is examined independently from any ‘making drink’ activity.

$$\phi_5 = AG(PanOnStove \rightarrow (AF(PastaReady) \rightarrow EX(FinishEating \rightarrow Exiting)))$$

Once again, the true outcome of this property shows that a user can successfully complete that activity and then stop interacting with the system by exiting from the kitchen.

The last two cases that will be model checked involve the non-completion of activities<sup>3</sup>. These activities are related to the preparation of tea or coffee.

The following properties examine the non-completion of the activities ‘making a tea’ and ‘making a coffee’ by the user. It is assumed that these activities follow the successful preparation of the meal. The CTL proposition for those properties are presented below:

$$\phi_6 = EG(PastaReady \rightarrow EX((\neg TeaReady \wedge FinishEating) \rightarrow AF(Exiting)))$$

$$\phi_7 = EG(PastaReady \rightarrow EX((\neg CoffeeReady \wedge FinishEating) \rightarrow AF(Exiting)))$$

The verification outcome of these properties indicates that these are potential system interactions with the users, which can be considered as abnormal behaviours that could point to a worrying situation for the elderly users. This conclusion is influenced by real-life situations, where under normal conditions people complete these tasks without facing problems.

Other interesting properties examined in relation to the abnormal behaviour that can take place in the smart kitchen environment are: (i) opening and closing of cupboards without taking ingredients or utensils, (ii) opening and closing fridge door without getting the milk and (iii) getting ingredients required for making a drink but not using them. Example properties of these behaviours are presented below:

$$\phi_8 = EF(OpenCupBD1 \rightarrow EX(\neg(GotTeaBag \wedge GotCoffeeNeeded \wedge GotSugarNeeded) \wedge MainKitchArea))$$

$$\phi_9 = EG(MainKitchArea \rightarrow EF(OpenDoor \rightarrow EX(\neg GotMilkNeeded \wedge MainKitchArea)))$$

The examination of all the above properties has shown that these behaviours are likely to occur throughout the system interaction with the users. Therefore, the repetitive occurrence of such behaviours should be considered as ‘suspicious’ and should require a closer

<sup>3</sup>The verification of last case scenario, in Section V, is not presented as it can be considered as part of the previously examined cases.

observation of the elder users' activities. It is also worth noting that the identification of normal and abnormal behaviour can be further examined by considering also the concurrent behaviour of the model. Therefore, for the examination of ADLs that are performed concurrently by the users exhibiting normal or abnormal behaviours, the following two properties are investigated:

$$\phi_{10} = EF(((|Cup| = 1 \rightarrow |CoffeeReady| = 1) \wedge (|PanOnStove| = 1 \rightarrow |PastaReady| = 1)) \rightarrow EX((|AtTableDrinking| = 1 \wedge |FinishEating| = 1) \rightarrow |Exiting| = 2))$$

$$\phi_{11} = EF(((|Cup| = 0 \rightarrow |CoffeeReady| = 0) \wedge (|PanOnStove| = 1 \rightarrow |PastaReady| = 1)) \rightarrow EX((|AtTableDrinking| = 0 \wedge |FinishEating| = 1) \rightarrow |Exiting| = 2))$$

The property  $\phi_{10}$  examines the successful completion of two concurrent activities performed by the two users of the system. This case corresponds to scenario nine that is presented in Section V demonstrating normal concurrent interactions between the users and the smart environment. On the contrary, property  $\phi_{11}$  examines again two concurrent activities, but in this case, the 'making coffee' and 'making meal' activities are not completed and completed respectively. For both properties, the verification result is true satisfying that these activities can be conducted simultaneously representing the detection of both the normal and abnormal behaviours of the users.

Finally, further properties and scenarios can be checked with respect to the detection of normal or abnormal behaviour considering both the sequential and concurrent execution of the ADLs in the developed model.

## VII. CONCLUSIONS

This paper presented the use of Petri nets to model three ADLs for the identification of the completion or non-completion of the activities as indicators of normal or abnormal behaviour, respectively. The three kitchen ADLs considered (preparing tea, preparing coffee and preparing pasta) are deemed to have many occurrences during a typical day of an elderly person, hence their relevance. Furthermore, the description of the sensor system of the kitchen ADLs, the developed Petri net model and its verification results were presented. From processing the data collected from the sensors, it was noticed that the activities could be accomplished by performing certain sequences of actions. Following this observation, the model explicitly represents these sequences of actions, as is described in the paper. From the Petri net model and the results obtained, we arrived at the following conclusions:

- 1) The behaviour of the Petri net modelling of the kitchen ADLs presented is reliable and reflects real world behaviour. Thus, the user is not restricted by the system, in this case the smart kitchen, to follow his/her initiative to perform any desired activity. This freedom of action enables the system for the detection of normal or abnormal behaviour.
- 2) The modularity and compositionality of the Petri net model of the kitchen ADLs allows for the addition or removal of new activities to the sensor system without drastic changes to the entire model.
- 3) The model represents the sequential (one user) or concurrent (multiple users) behaviour of the smart kitchen just as it typically occurs in the real world.
- 4) Different type of sensors can be modelled and added to the system using Petri nets.

Future work will consider the following elements added to the sensor system for kitchen ADLs and respective Petri net model: (i) different ADLs, (ii) different type of sensors, and (iii) more users. Specific cases of normal and abnormal behaviour will be considered.

## REFERENCES

- [1] M. A. Blatke, M. Heiner, and W. Marwan, "Tutorial - Petri Nets in Systems Biology", Technical report, Otto von Guericke University Magdeburg, Magdeburg Centre for Systems Biology, 2011.
- [2] D. Bjørner, C.B. Jones, M. Mac an Aircinnigh and E.J. Neuhold, "VBM '87: VDM - A formal method at work, In Proceedings of the VDM-Europe Symposium, 1987.
- [3] L. Chen, C.D. Nugent, and H. Want, "A Knowledge-Driven Approach to Activity Recognition in Smart Homes", IEEE Transactions on Knowledge and Data Engineering, IEEE, Vol. 24, No. 6, pp. 961-974, 2012.
- [4] L. Chen, J. Hoey, C.D. Nugent, D.J. Cook, and Z. Yu, "Sensor-Based Activity Recognition", IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), IEEE, Vol. 42, No. 6, pp. 790-808, 2012.
- [5] E. Clarke, O. Grumberg, and D. Long, "Verification tools for finite-state concurrent systems", In A Decade of Concurrency Reflections and Perspectives, Lecture Notes in Computer Science, Vol. 803, Springer Berlin Heidelberg, pp. 124-175, 1994.
- [6] M. Ermes, J. Pärkkä, J. Mäntyjärvi, and I. Korhonen, "Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions", IEEE Transactions on Information Technology in Biomedicine, Vol. 12, No. 1, pp. 20-26, 2008.
- [7] Cl. Girault and R. Valk, "Petri Nets for System Engineering: A Guide to Modeling, Verification, and Applications", Springer-Verlag New York, 2001.
- [8] M. Heiner, M. Herajy, F. Liu, C. Rohr, and M. Schwarick. "Snoopy a unifying petri net tool", In Application and Theory of Petri Nets, volume 7347 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, pp. 398-407, 2012.
- [9] Y. Hong, I. Kim, S. Ahn, and H. Kim. "Activity recognition using wearable sensors for elder care", Second International Conference on Future Generation Communication and Networking, IEEE, vol. 2, pp. 302-305, 2008.
- [10] K. Jensen, "Coloured Petri nets: basic concepts, analysis methods and practical use", Vol. 1, Springer Science & Business Media, 1997.
- [11] Y. Jing, M. Eastwood, B. Tan, A. Konios, A. Hamid and M. Collinson, "An Intelligent Well-being Monitoring System for Residents in Extra Care Homes", In Proceeding of In Proceedings of IML conference, 2017 (IML 2017).
- [12] A. Konios, "Modelling and Verication of Ambient System Using Petri Nets", Ph.D. Dissertation, Newcastle University, 2015.
- [13] A. Konios and M. Pietkiewicz-Koutny, "Modelling Ambient Systems with Coloured Petri Nets", Technical Report, School of Computing Science, Newcastle University, 2015.
- [14] S. Merz, "Model checking: A tutorial overview", In Modeling and Verification of Parallel Processes, pp. 338, 2001.
- [15] X. Niu and Z. Wang, "A Smart Home Context-aware Model Based on UML and Colored Petri Net", International Journal of Smart Home, Vol. 10, No. 1, pp. 101-114, 2016.
- [16] A.K. Nabih, M.M. Goma, H.S. Osman and G.M. Aly, "Modeling, Simulation and Control of Smart Homes Using Petri Nets", International Journal of Smart Home, Vol. 5, No. 3, pp. 1-14, 2011.
- [17] C. A. Petri, "Kommunikation mit Automaten", Ph.D. Dissertation, Bonn: Institut für Instrumentelle Mathematik, 1962.
- [18] J. Rafferty, J. Synnott, A. Ennis, C. Nugent, I. McChesney, and I. Cleland. "SensorCentral: A Research Oriented, Device Agnostic, Sensor Data Platform", International Conference on Ubiquitous Computing and Ambient Intelligence, Springer, pp. 97-108, 2017.
- [19] D. Sangiorgi and D. Walker, "PI-Calculus: A Theory of Mobile Processes", Cambridge University Press, 2001.
- [20] E. Tapia, S. Intille, and K. Larson, "Activity Recognition in the Home Setting Using Simple and Ubiquitous Sensors", Pervasive Computing, Springer, Vol. 4, pp. 158-175, 2004.
- [21] L. Wang, T. Gu, X. Tao, H. Chen, and J. Lu, "Recognizing multi-user activities using wearable sensors in a smart home", Pervasive and Mobile Computing, Elsevier, Vol. 7, No. 3, pp. 287-298, 2011.
- [22] J. Woodcock, P.G. Larsen, J. Bicarregui and J. Fitzgerald, "Formal methods: Practice and experience", ACM Computing Surveys, 41(4):1-36, 2009.