Anomaly Detection in the Elderly Daily Behavior

Parvaneh Parvin
CNR-ISTI, HIIS Laboratory
University of Pisa
Pisa, Italy
parvaneh.parvin@di.unipi.it

Fabio Paternò
CNR-ISTI, HIIS Laboratory
Pisa, Italy
fabio.paterno@isti.cnr.it

Stefano Chessa
CNR-ISTI, WN Laboratory
University of Pisa
Pisa, Italy
stefano.chessa@unipi.it

***Abstract*—The increasing availability of sensors and intelligent objects enables new functionalities and services. In the Ambient Assisted Living (AAL) domain, such technologies can be used for monitoring and reasoning about the older people behavior to detect possible anomalous situations, which could be a sign of the next onset of chronic illness or initial physical and cognitive decline. We propose an approach to detecting abnormal behavior by developing a profiling strategy (in which task models specify the normal behavior), which can also work in case of rare anomaly data. Events corresponding to the user behavior is detecting by a middleware software(Context Manager). Afterward, our algorithm compares the planned and actual behavior to identify if any deviation occurred and also defines to which category the anomaly belongs. The resulting environment should be able to generate multi-modal actions (i.e alarms, reminders) based on detected anomalous behavior, aiming to provide useful support to improve older people well-being.**

***Keywords*-Elderly Behavior Analysis, Deviations in Task performance, Ambient Assisted Living.**

I. Introduction

As people become older, they depend more heavily upon outside support for health assessment, caregivers, and medical care. With the growing age, a person may encounter with different impairments physically and mentally such as vision problems, cognitive problems (dementia) etc. Nowadays, ambient intelligence technologies are increasingly used as means to support everyday living of older people individuals by monitoring them through a plenty of sensors and interactive devices. In the Ambient Assisted Living (AAL) scenarios, the analysis of the data so gathered can be useful for identifying the current state and activity of people. It helps to provide them a safe and secure ambient to live independently in their homes rather than through more expensive hospitalization solutions and give them the ability to perform basic activities of daily living (e.g. bathing, dressing, food preparation, taking medicines), on which the ability of a person to live independently is assessed. In today's world, most of the data are streaming and time-series. Detecting the anomalous behavior of the user by using these data gives useful information about risky situations. By increasing the number of older people, the streaming data get bigger and bigger and detecting anomalies in a massive streaming data is difficult because it needs sensors, smart devices, detectors to analyze the data in a real-time as soon as an event occurs. In addition, detecting anomalous behavior in real-time requires fast and precise algorithm as well [1]. Though there are many studies addressing anomaly detection in user behavior, there have been few contributions point to the issue of detecting possible deviations from expected users routines. In one hand, remote monitoring of vital signs and daily activities of the older people is highly critical because changes in their routine can influence early detection of illness. On the other hand, the identification of anomalous patterns can be highly valuable for experts and caregivers to react quickly when a sudden change in medical parameters occurs. Furthermore, deviations could become manifest in the different ways based on the activity, place, time and the duration that the activity is carried out compared to the expected one. Also, the severity degree of the activity should be considered, just as, the consequence of not taking a pill could be more severe than just forgetting to take a shower. In addition, time plays a serious role, because some anomalies need longer time-span to consider as a serious, risky situation and some are detectable immediately. For example, going to the bathroom is not a risky situation by itself but if the older people repeat this activity too frequently in a day, it can be an initial symptom of health problem. Our approach employs task model specifications to plan the user daily routine and associate the elementary tasks in a task model to the real events logged from the current context. Further, by using the anomaly detection algorithm, we compare the current users behavior to expected one defined in the task model to detect the deviation if it is any. The system can also distinguish the different types of anomalies and triggers a suitable action addressing the concerned abnormality, accordingly.

II. Related Work

Many studies have been conducted to define the needs of older people concerning a Remote health care system able to detect anomalous behavior and help them in their daily life. A survey [2] about anomaly detection approaches in smart homes basically identifies two ways to detect behavioral changes: profiling and discriminating. The former is modeling normal behavior and considering values that do not comply with the model as anomalies while the latter learn anomaly data from historical data and search for similar patterns from incoming data to find anomalies. Monekosso and Remagnino [3] proposed a model-based behavior analysis system. They using Hidden Markov Models (HMM) to model user behavior from sensed data. They identified activities (i.e. cooking,

2472-7571/18/$31.00 ©2018 IEEE
DOI 10.1109/IE.2018.00025

eating, etc.) corresponding to the pattern of events which gathered by sensors. The system is able to recognize the anomalies (e.g. repeatable patterns) against a normal behavior. But unlike our work, the detected anomalies in their work need further examination of a domain expert to precisely indicate the nature of the anomaly. Ranges of machine learning techniques have been used for anomaly detection such as clustering [4], Support Vector Machines [5], Neural Networks [6], HMM [3] and etc. However, they have some drawbacks. The SVM method requires pre-classified data to characterize the range of possible anomalous/non-anomalous data that may be encountered. The HMM is complex when there is a large set of transitions of all possible states (e.g. patient with the cognitive problems). The Neural network approach can be slow, expensive to train and has a complex network architecture, and, if there is a lack of data or no learnable function, it does not provide a valid solution.

On the other hand, task analytic methods have been used in different studies to demonstrate the regulating human behavior [7]. The resulting models are often hierarchical and represent the mental and physical activities human operators performed to reach their goals. Researchers have developed task analytic modeling systems such as: AMBOSS [8]; Enhanced Operator Function (EOFM) [9]; ConcurTaskTrees (CTT) [10]. In [11], an extension of CTT-notation, a hierarchical task model was presented to specify the probabilities of the next possible action during the execution that allows a transformation into Markov Models. The system, having recognized the current user state, can then make use of the model in order to adjust the probabilities of the next possible state. Task models are able to specify the same information in a more compact way. The transformation algorithm allows generating an initial version of a Markov Model from a task model that can be adapted in further development.

In this area, Pollack, describes Autominder [12], a system intended to help older adults adapt to cognitive decline and continue a satisfactory performance of routine activities by providing adaptive, personalized reminders of daily living activities. Their solution uses a range of intelligent techniques to model individuals daily plans, to observe and reason about the execution of those plans, and to make decisions about whether and when it is most appropriate to issue reminders. The effect of a deviation from the specified plan is the generation of a reminder. Nevertheless, this system is not able to distinguish the type of cognitive errors carried out by the user which is important for adapting the reminders strategy. While, our approach presents a system able to specify each task and its anomaly grade and a wider range of actions depending on the detected anomaly. Paganelli et al., has considered task models to compare expected user behavior and actual user interactions with web applications in order to identify possible usability issues [13] but in our approach we considered events occurring in the user context which is associated with ubiquitous sensors that appeared anomalous has more severe consequences than the one for website usability issue. Meanwhile, our system also analyzes the identification of suitable actions to perform in real-time to cope with the identified deviation.

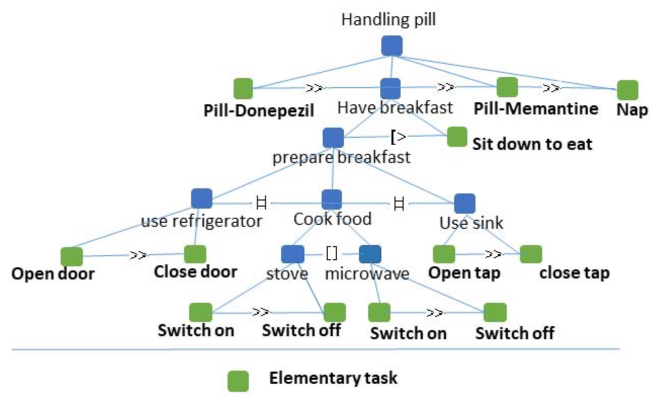

Fig. 1. The task model of the example.

III. EXAMPLE USE CASE

We considered an example to highlight the possible task-related anomalies in elderly behavior. The task model of this use case is shown in “Fig. 1” and describes a part of expected morning routine of a user (Sara) with Dementia. Sara should take her medicine (Donepezil) in the morning before having her meal. Then, she should make the breakfast (take food from the refrigerator, cook/reheat food) and only after the breakfast can take another pill (Memantine), and usually, after that, she takes a nap for 20 minutes. From this use case, it is possible to derive various ways in which Saras activities can be performed in a different manner from the expected one. One of the most critical deviations involves medicine intake: Confusing medication and taking two different drugs together that magnify each other’s potential side effects, take the same medicine twice, taking medication in a wrong order or forget to take the medicine. Thus, deviations involving critical tasks, such as medicine intake need short time span to be analyzed and should receive prompt attention. However, some tasks (e.g. taking nap) need longer timespan (the user sleeps more and wakes up late, considering a week) to be interpreted as a risky situation.

IV. METHOD

Our method is composed of several steps, which are described in the following:

Set up a suitable task model for the analysis. The task model describes the planned activities from the users viewpoint should be created with the collaboration of caregiver (or even the older people themselves), who have an intimate knowledge of the needs of their care receivers. To specify task models, we used CTT notation [10] which allows focusing on the various activities modeled in a hierarchical manner. A task model is composed of several tasks which are connected by various temporal operators such as enabling ($\gg$), disabling ($[>$), interruption ($|>$), choice ($[]$), iteration ($[\overset{*}{t}]$) , concurrency ($|||$) , optionality ($[t]$) and order independency ($|=|$). Each task is an activity that users do to achieve a goal and expresses one or multiple events in a context.

Associate events occurred in the context with elementary tasks in the task model. After creating a task model, we create a file containing the mapping between the elementary tasks in the task model and the events in the current context. It is worth mentioning, not all the tasks are corresponding to a single event. In some cases, an elementary task is equivalent to the multiple events (e.g. having breakfast task depending on the available sensors could be: user location= kitchen AND oven sensor=on AND time¡09:00). Thus, a task is considered complete when all the events associated with the task has been detected.

Log relevant contextual events. At the third step, the purpose is to log the events (i.e events related to the elementary tasks contained in the task model) that occur in the older people current context and associating them with timestamps. Indeed, there is a Deviation Analysis module (in Section V) which subscribes to the Context Manager to receive such events and then saves them in a specific data structure for further analysis.

Detecting anomalies by comparing the actual and the expected user behavior. To detect anomalies in our discrete sequences, we used the sequence-based anomaly detection techniques that compare the test data set with the task model (train set) in order to detect the anomalies. Thereby, we developed an anomaly detection algorithm which compares the set of tasks enabled in the task model (according to the current execution state of the model) and the task associated with the real events sequence occurred in the context: if the latter one does not belong to the first one, then a deviation occurs. Several types of deviations can be identified as a result of this comparison. In particular, since users activities can be described in terms of a set of task attributes (e.g. temporal relationships, location, time), it is possible to identify a number of corresponding types of task-related anomalies affecting older people that we categorized them as below:

- *Less:* A task that was expected to be carried out, has not been performed;
- *More:* A task has been performed repeatedly;
- *Difference:* Expected task has been performed but in a wrong manner (in terms of temporal order, time, location).

V. ARCHITECTURE

Fig. 2 shows the architecture of the system which consists of two main phases. In the first phase, there is an association tool that receives as an input the user task model and as an output, it gives an association file containing the mappings between the elementary tasks in the task model and the events in the current context. It is worth pointing out, not all the tasks coincide with one event, in some cases, a single task is associated with multiple events. In this case, a task will be considered complete if all the events associated with the task will occur in the context. The second phase deals with the detection of user anomalies behavior, which is composed of four sub-modules: the Deviation Analysis, the Simulator, the Persuasion Module, and the Action Generator.

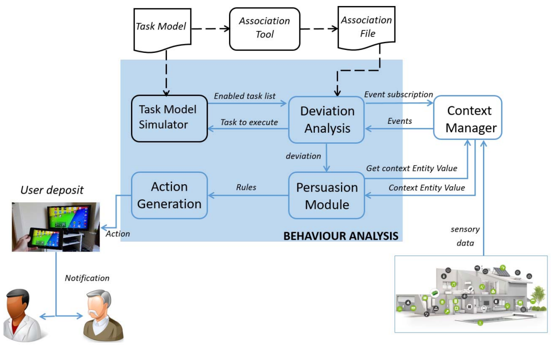

Fig. 2. The Architecture of the Solution.

Deviation Analysis module takes the association file related to the user task model and subscribes it to the Context Manager in order to be notified when the events specified in the association file occur in the context. The Context Manager [14] is a software module able to detect the current values of a wide range of variables (related to the user, environment, technology, etc.) and to inform other architectural modules about relevant changes in such values. So, Deviation Analysis module as an input gets three elements: i) the association file; ii) the event lastly occurred in the current context (provided by Context Manager); iii) the enabled task sets (from the simulator). The simulator module gets in input the CTT task model starts from the initial task and depending on the task temporal operators, hands out the list of the tasks currently enabled after the execution of the given task. This process continues until it arrives at the final task. Thus, the basic goal of the Simulator module is to provide the Deviation Analysis with the information needed to decide whether the sequence of events detected by the Context Manager can be translated in a path of elementary tasks representing the expected behavior of the user. Nevertheless, The behavior of the anomaly detection algorithm is the following one:

It takes as an input the event sequence happened provided by Context Manager, it retrieves in the association file the corresponding task model, it uploads the file to the simulator and checks whether each task from the event sequence belongs to the set of enabled tasks provided by the Simulator module (e.g. if taking "pill-Memantine" is enabled after "having a breakfast"). If this is the case, it means that the received event was an expected one (according to the task model), then the process iterates in a similar manner for the next task in the received event sequence. Otherwise, the algorithm marks the sequence as an abnormal one and starts to define the anomaly type by considering the proper time span and the criticality level of the anomalous task. As an output, the algorithm marks each anomalous task as a *Less* (missing tasks), *More* (more tasks than expected have been performed) or *Difference* (tasks that have been performed differently from what the designer intended in the case of time and

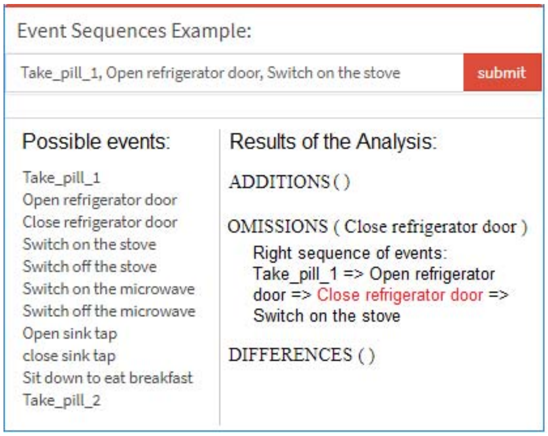

Fig. 3. An example of analysis result.

order of performance). Later, the Deviation Analysis sends the task along with its anomaly type to the Persuasion Module. The Persuasion Module uses the data collected by deviation analysis, the critical level of the involved task and also refers to the Context Manager to reason about the current behavior of the user to decide whether the triggered action should be persuasive. Consequently, it defines the motivation techniques and rules that can be applied to reinforce or change the anomalous behavior aiming to improve the elder well-being and passes these rules to the Action Generation module. Finally, when the Action Generation module received such rules (which define the most suitable ways to render the actions identified for addressing the diagnosed deviation), by considering relevant contextual aspects, generates a suitable action such as modifying the user interface, change the state of appliance, activating some functionality, generating alarms and reminders or provide persuasive suggestions to encourage users for changing their behavior. As an initial result, we developed an application (see Fig. 3) to detect the deviation by comparing the expected user behavior (i.e task model) with the real one. In the example, the task model is still the one shown in Fig. 1 and the only deviation identified was a Less deviation (i.e. a task was not done), as the task "Close refrigerator door" has not been carried out.

VI. Conclusion

We have presented a method to detect abnormal behavior of older people, which is important in AAL scenarios because these anomalies may represent early signs of the onset of health-related issues. Our solution exploits task model specifications and compares the expected behavior expressed in such models with the sequence of events generated by real user behavior. As a preliminary result, the detection is done in an offline manner (i.e. the entire event sequence is available at the end of each day) and by using our anomaly detection algorithm, each event sequence is labeled either normal or abnormal and in the case of abnormalities, the deviation type (*Less, More, Difference*) has been defined. For achieving the result, we had to change manually some properties of context model (i.e. user, environment, technology, social) in order to receive the information based on the user activity. Meanwhile, we plan to extend the analysis by considering other task-related aspects/attributes (e.g. task time and duration) and to empirically validate our approach by conducting a field experiment for detection of anomalies involving real data and also expand our approach to the real-time anomaly detection. Hence, the anomaly detection method must be as fast as possible to ensure that detection keeps up with the rate of data collection. Obstacles to achieving anomaly detection in real-time include the large volume of data associated with user behavior and the nature of that data [15]. Using our method, we overcome this obstacle by subscribing to the Context Manager in order to be notified when the events specified in the association file occur in the context. This way, we just receive the relevant events from the Context Manager.

References

[1] Z. Hasani, *Robust anomaly detection algorithms for real-time big data: Comparison of algorithms*. In Embedded Computing (MECO), IEEE 6th Mediterranean Conference, 2017, pp. 1–6.

[2] U. Bakar, H. Ghayvat, S. Hasanm, S. Mukhopadhyay, *Activity and anomaly detection in smart home: A survey*. In: Next Generation Sensors and Systems, Springer International Publishing, 2016, pp. 191–220.

[3] Monekosso, N. Dorothy, and P. Remagnino. *Behavior analysis for assisted living*. IEEE Transactions on Automation science and Engineering, 7.4 (2010), pp. 879–886.

[4] B. T. Fine, *Unsupervised anomaly detection with minimal sensing*. Proceedings of the 47th Annual Southeast Regional Conference, ACM, 2009. pp. 60.

[5] D. J.C. Vikramaditya, R. Jakkula, D. J. Cook. *Detecting Anomalous Sensor Events in Smart Home Data for Enhancing the Living Experience*. Artificial intelligence and smarter living, 2011, 11.201: 1.

[6] G. Poojitha, K. Naveen Kumar, and P. Jayarami Reddy. *Intrusion detection using artificial neural network," Computing Communication and Networking Technologies (ICCCNT)*. International Conference on. IEEE, 2010.

[7] B. Kirwan, LK. Ainsworth, *A guide to task analysis: the task analysis working group*. , CRC press, 1992.

[8] M. Giese, T. Mistrzyk, A. Pfau, G. Szwillus, M. Detten, *AMBOSS: a task modeling approach for safety-critical systems*. In: Engineering Interactive Systems. Springer, Berlin, Heidelberg, 2008. pp. 98–109.

[9] M. L. Bolton, R. I. Siminiceanu, E. I. Bass, *A systematic approach to model checking humanautomation interaction using task analytic models*. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 2011, 41.5: pp. 961-976.

[10] G. Mori, F. Paternò, C. Santoro, *CTTE: support for developing and analyzing task models for interactive system design*. IEEE Transactions on software engineering, 2002, 28.8: pp. 797–813.

[11] M. Giersich, P. Forbrig, G. Fuchs, T. Kirste, D. Reichart, H. Schumann, J. Jacko, *Towards an integrated approach for task modeling and human behavior recognition*. In Human-Computer Interaction. HCII, Heidelberg:Springer Verlag, 2007, pp. 1109–1118.

[12] M. E. Pollack,*Autominder: A case study of assistive technology for elders with cognitive impairment*. Generations 30, no. 2 (2006): pp. 67–69.

[13] L. Paganelli, F. Paternò, *Tools for remote usability evaluation of Web applications through browser logs and task models*. Behavior Research Methods, 2003, 35.3: pp. 369–378.

[14] G. Ghiani, M. Manca, , F. Paternò , C. Santoro, *Personalization of context-dependent applications through trigger-action rules* ACM Transactions on Computer-Human Interaction (TOCHI), 2017, 24.2: 14.

[15] L. Lankewicz, M. Benard, *Real-time anomaly detection using a non-parametric pattern recognition approach* IEEE, Computer Security Applications Conference, 1991. Proceedings., Seventh Annual, pp. 80–89.