Monitoring Domestic Activities with Temporal Constraints and Components

M. CIRILLO [a,1], F. LANZELLOTTO [b,2], F. PECORA [a] and A. SAFFIOTTI [a]

[a] *AASS Mobile Robotics Lab, Örebro University, Sweden*

[b] *Dept. of Informatics and Automation, University Roma-3, Italy*

Abstract. Intelligent environments are increasingly rich in ubiquitous sensing capabilities that can be leveraged to know which actions a user is engaged in at any given moment in time. The ability of an intelligent environment to recognize a high-level plan of activities performed by the user in a smart home would allow to construct proactive services, such as reminding, forecasting and providing timely physical support. This article proposes an approach to human activity recognition based on temporal planning. The approach leverages on one hand the ubiquitous sensors provided by the PEIS-Home, a sensor-rich intelligent environment, and, on the other hand, the temporal representation and reasoning capabilities of OMPS, a constraint-based temporal planning and scheduling framework.

Keywords. Intelligent environments, activity recognition, ubiquitous sensors, planning, temporal reasoning

Introduction

In this paper we tackle the problem of recognizing patterns of activities carried out by a human being within an intelligent home environment equipped with pervasive and heterogeneous sensors. This problem has gained the attention of a number of disciplines within the artificial intelligence community, as the ability to automatically recognize activities is a key capability for building intelligent environments that provide personalized and effective support services. In order to achieve this, an intelligent system must employ non-trivial and temporally contextualized knowledge regarding the state of the user [2]. For instance, if a smart home could recognize that the human user is cooking, it could avoid cleaning the dining room until the subsequent dining activity is over.

Current approaches to this problem can be roughly categorized as *data-driven* or *knowledge-driven*. In data-driven approaches, models of human behavior are learned from large volumes of data over time. Notable examples of such approaches employ Hidden Markov Models (HMMs) for learning sequences of sensor observations with given transition probabilities. Instantiated within the domestic environment setting, these techniques have been leveraged to infer human activities from RFID-tagged object use [12] as well as vision-based observations [17,14].

[1] Corresponding author: Örebro University, 701 82 Sweden. E-mail: marcello.cirillo@aass.oru.se. This work has been partially supported by CUGS (Swedish national computer science graduate school) and by the Swedish KK foundation.

[2] Work performed while this author was at the AASS Mobile Robotics Lab, Örebro University, Sweden.

In knowledge-driven approaches, patterns of observations are modeled from first principles rather than learned. Abductive processes are employed instead, whereby sensor data is explained by hypothesizing the occurrence of specific human activities.[3] Examples include the work of Goultiaeva and Lespérance [8], where the Situation Calculus is used to specify very rich plans, as well as approaches based on ontologies [10] and temporal reasoning approaches in which rich temporal representations are employed to model the conditions under which patterns of human activities occur [13,9].

Data- and knowledge-driven approaches have complementary strengths and weaknesses. One advantage of the former is the ability to learn patterns of human behavior, rather than having to model them explicitly. This advantage comes at the price of poor scalability and inherent difficulty of providing common-sense knowledge for classifying overwhelming amounts of data [17]. In this sense, knowledge-based approaches can provide a powerful means to express such information, which can then be employed by an abductive process that essentially "attaches a meaning" to the observed data. Indeed, the current literature points to the fact that the two strategies are complementary in scope: data-driven approaches provide an effective way to recognize elementary activities from large amounts of continuous data; conversely, knowledge-driven approaches are useful when the criteria for recognizing human activities are given by complex but general rules that are clearly identifiable. While the ranges of applicability of the two strategies clearly overlap, knowledge-driven approaches have been less explored in literature. We argue that such strategies retain an advantage in domains where the emphasis lies not so much in recognizing elementary actions, rather where knowledge modeled from first principles can be used to fuse heterogeneous sensor data to obtain higher-level activity inference.

In this paper we present a knowledge-driven approach based on a rich temporal representation and on a planning framework. The focus of this paper is on the knowledge representation aspect of the architecture. We show how the use of a component-based domain representation grounded on Allen's Interval Algebra [1] is employed in conjunction with a constraint-based planning and scheduling framework called OMPS [7] to obtain an abductive process for activity recognition. The use of this architecture is demonstrated within a real intelligent environment, namely the PEIS-Home [15], a testbed environment in which robotic, sensory and intelligent software components are combined to obtain a service providing environment for the future home.

1. Background: the OMPS Framework

The Open Multi-component Planner and Scheduler (OMPS) [7] is a constraint-based planning and scheduling software framework grounded on the notion of *component*. OMPS has been used to develop a variety of decision support tools, ranging from highly-specialized space mission planning software to classical planning frameworks [4]. A complete description of OMPS is outside the scope of this paper. We focus here on OMPS' representation and reasoning capabilities which are of interest for the problem of deducing patterns of human activities from sensor readings in an intelligent environment.

A component is an element of a domain theory which represents a logical or physical entity. Components model parts of the real world that are relevant for a specific decisional process, such as complex physical systems or their parts. Components can be used to represent, for example: a robot which can navigate the environment and grasp objects; a simple luminosity sensor which can estimate the lighting conditions in a room; or an

[3] An approach similar to Shanahan's work [16] on inferring information on a robot's environment.

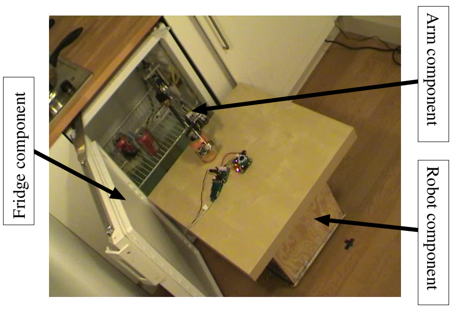
autonomous refrigerator which can open and close its door and determine which products are stored in it.

Components are the basic elements used to describe the domain of a specific problem in OMPS. Simply put, an automated reasoning functionality developed in OMPS consists in a procedure for taking *decisions* on components. Decisions essentially describe an assertion on the possible evolutions in time of a component. For instance, a decision on the fridge component described above could be to open its door no earlier than time instant 30 and no later than time instant 40; or a decision on the robot above could assert that the robot is to be in the state "localized" using its laser for self localization. More precisely, a decision is an assertion on the value of a component in a given flexible time interval, i.e., a pair $\langle v, [I_s, I_e]\rangle$, where the nature of the value v depends on the specific component and I_s, I_e represent, respectively, an interval of admissibility of the start and end times of the decision. In the fridge example, assuming the door takes five seconds to open, the flexible interval is $[I_s = [30, 40], I_e = [34, 44]]$, while in the second example no temporal requirement is imposed on robot localization (i.e., $I_s = I_e = [0, \infty)$).

The core intuition behind OMPS is the fact that decisions on certain components may entail the need to assert decisions on other components. For instance, the decision to open the fridge door may *require* that the robot is docked to the fridge so that it can grasp an object inside. Such dependencies among component decisions are captured in a domain theory through what are called *synchronizations*. A synchronization states the requirements entailed by a decision on other components. These requirements consist in decisions that are bound to the "requiring" decision by *temporal constraints*. OMPS provides thirteen types of temporal constraints, each type corresponding to one of the relations in Allen's Interval Algebra [1].

Arm : **place_can_on_robot**
DURING Robot : **docked**
DURING Fridge : **open**
CONTAINS Arm : **grasp_can**
CONTAINS Arm : **release_can**

(a)



(b)

Figure 1. One synchronization in a possible domestic robot planning domain (a), and the corresponding real components available in the PEIS-Home (b).

For example, the synchronization in figure 1(a) states that asserting **place_can_on_robot** on a component representing a robotic arm inside the fridge entails that other decisions should be asserted on the robot (which should be docked to the fridge DURING the course of the operation), on the fridge door (which should be open DURING the operation) and on the arm itself (the **place_can_on_robot** operation should CONTAIN the operations of grasping and releasing the can). Other synchronizations in the domain can be employed to model other requirements among components, e.g., the need to schedule the **grasp_can** operation BEFORE the **release_can** operation.

Overall, the fundamental building blocks of OMPS that are of interest for this article are components, decisions and temporal constraints. Together, decisions and temporal constraints asserted on components are maintained in a *decision network* (DN), that is at all times kept consistent through *temporal propagation*. This ensures that the temporal intervals underlying the decisions are kept consistent with respect to the temporal constraints, while decisions are anchored flexibly in time. In other words, adding a temporal constraint to the DN will either result in the calculation of updated *bounds* for the intervals I_s, I_e for all decisions, or in a propagation failure, indicating that the added constraint or decision is not admissible (e.g., adding **A** BEFORE **B** when the DN contains constraints **B** BEFORE **C** and **C** BEFORE **A**). Temporal constraint propagation is a polynomial time operation, as it is based on a Simple Temporal Network [6].

2. From Planning in Space to Activity Recognition at Home

Although general, OMPS was built with space applications in mind [5]. A direct consequence of this fact is that components represent *controllable* entities, i.e., the semantics of the term "component decision" is that of an assertion that prescribes an operation on a component. For instance, asserting a decision $\langle \mathbf{open}, [[163, 163], [164, \infty)] \rangle$ on the component representing an autonomous fridge models the operational request of the fridge to open its door at time 163 and that the operation should last at least one time unit. This mindset is appropriate for building planning/scheduling systems whose purpose is to deduce operational plans for controllable systems. Indeed, OMPS can be seen as a symbolic framework for developing controllers for complex systems (such as space mission planning processes). This transfers very directly into intelligent environment settings such as the PEIS-Home if the aim is to control pervasive actuators, or robots, or intelligent services. However, we address here the complementary problem of employing the framework for automatically deducing the activities carried out by a partially observable, and certainly not controllable component, namely the *human user*. This deductive process is based on the values of equally non-controllable, but directly observable components, namely *sensors*.

2.1. The Human User

The human inhabitant of the environment can be modeled as a state variable, whose values correspond to symbols representing the various activities that the person can perform in the home. Synchronizations are used to model the conditions under which specific human activities are recognized. For instance, the requirements expressed in synchronization {Human : **cooking** EQUALS Stove : **on**, DURING Location : **kitchen**} models the (very simple) criteria for recognizing the cooking activity: this activity requires that the user is in the kitchen and it lasts as long as the stove is turned on. The activity recognition functionality is thus obtained by iteratively (1) updating the DN with values representing the observations of all sensors and (2) re-planning using a planning procedure which is very similar to that illustrated in [7]. As far as the discussion in this paper is concerned, it is sufficient to say that the procedure performs a search in the space of DNs for a set of synchronizations that is applicable given the current state of the DN. For instance, suppose the current time is 32, and that the DN contains two decisions: $d_{\text{Location}} = \langle \mathbf{kitchen}, [[2, 2], [32, \infty)] \rangle$ on component Location, representing that the person-localization sensor has observed the human user as being in the **kitchen** since time 2, and $d_{\text{Stove}} = \langle \mathbf{on}, [[10, 10], [32, \infty)] \rangle$ on component Stove, representing that the stove has been observed to be in state **on** since time 10. Then the activity recognition process would deduce, based on the synchronization above, a decision $d_{\mathrm{Human}} = \langle \mathbf{cooking}, [[10, 10], [32, \infty)] \rangle$, representing the fact that human activity "cooking" has been recognized as taking place since time instant 10. It is important to notice that the recognized **cooking** decision is *flexibly bound* to the two sensor decisions through the constraints (d_{Human} EQUALS d_{Stove}) and (d_{Human} DURING d_{Location}). As a consequence, if the minimum durations of d_{Stove} and d_{Location} increase (as they would if at the next iteration the human being is still observed in the kitchen and the stove is still turned on) then the minimum duration of the deduced activity will also increase. This is achieved through temporal propagation of the constraints in the DN, therefore is done in polynomial time (in the number of decisions) at every iteration.

2.2. Sensors

The state variable metaphor is also appropriate for modeling the state of pervasive sensors in the environment. For instance, the sensor readings of a person localization system are modeled as values of a component "Location", a state variable whose values correspond to the symbols representing the possible locations of the person in the PEIS-Home.

In order to realize the interface between OMPS state variables and real-world sensors in the environment, a new OMPS component, the *sensor*, was developed. OMPS sensors extend the capabilities of the built-in state variable. An OMPS sensor is modeled in the domain for each physical sensor in the environment. Each OMPS sensor is provided with an interface to the physical sensor, as well as the capability to periodically update the DN with decisions and constraints that model the state of the physical sensor. Specifically, for each sensed value provided by a physical sensor, the corresponding OMPS sensor does the following: if the value was not sensed at the previous iteration, then a decision representing the observed value is added, as well as a temporal constraint that anchors its start time to the current clock reference, and any previous decision's end time is anchored to the current time; conversely, if the sensed value is equal to the one sensed at the previous iteration, the decision's duration is increased.

Unlike a state variable, a sensor is no longer a passive entity on which decisions are imposed, rather a process which can impose decisions on itself to reflect the reality it observes in its physical counterpart. The result is a temporal reasoning infrastructure in which a multitude of processes add decisions and constraints to the DN concurrently: each sensor is a process that adds decisions and constraints to represent the real-world observations coming from the environment; in turn, the current DN is manipulated by the high-level planning process used for activity recognition, which adds decisions and constraints modeling the current activity performed by the user. In a way, the DN is acting as a "blackboard" where decisions and constraints re-construct the observed reality as well as the current hypothesis on what the human being is doing. Decisions and constraints are added to the DN in real-time, i.e., as they are observed by the sensors or decided by the activity recognition process.

3. A Domain for Activity Recognition in the PEIS-Home

The goal of the test described in this section is to provide a demonstrative example of how we can leverage domain knowledge expressed as OMPS synchronizations to recognize human activities from sensor readings.

For the purposes of this test, we have instantiated three sensor components in OMPS. These components wrap three of the PEIS-Home's sensors, namely a stereo camera on the ceiling with a tracking system [11] for person localization (component Location), a

1) Human : **watchingTV**
EQUALS Location : **couch**

2) Human : **cooking**
EQUALS Stove : **on**
DURING Location : **kitchen**

3) Human : **eating**
EQUALS KTRfid : **dish**
DURING Location : **kitchenTable**

4) HumanAbstract : **meal**
STARTED-BY Human : **cooking**
FINISHED-BY Human : **eating**

5) HumanAbstract : **nap**
EQUALS Human : **watchingTV**
AFTER HumanAbstract : **meal**

Figure 2. Synchronizations defined in our domain for the Human and HumanAbstract components.

stove state sensor implemented with a LabJack (Stove), and an RFID reader mounted underneath the kitchen table (KTRfid) for sensing the presence of tagged kitchen utensils.

We model the user in the environment as two distinct components, that can assume different values. The synchronizations of the first component, *Human*, have constraints that are based on sensor values, while the ones of the second component, *HumanAbstract*, have constraints that are based on values of Human and HumanAbstract. This distinction allows us to reason at two different levels of abstraction, as will be explained in the following.

The synchronizations employed in this experiment are shown in figure 2. They describe temporal patterns of sensor values, and how these patterns should be interpreted with respect to the Human and HumanAbstract components. For instance, synchronization (1) means that asserting **watchingTV** on the Human component entails that the decision **couch** should be asserted on the component Location (that is based on sensor data). Moreover, the synchronization specifies the temporal constraints among decisions: the duration of the decision **watchingTV** should be EQUAL to the one of **couch**.

We also define two synchronizations for the component HumanAbstract (the name of this component emphasizes the higher lever of abstraction of the activities recognized): synchronization (5) is triggered if the user is perceived to perform activity **watchingTV** after a meal. In such a case, knowing the habits of the human, it is inferred that he or she is most probably taking a nap.

A full experiment with this domain in the PEIS-Home was run by connecting our framework to real sensors. The user (depicted in the insets in figure 3) performs a series of actions, one after the other, to verify the correct detection of activities. First he sits on the couch watching TV, then he moves to the kitchen and prepares a meal (that is, he enters the kitchen and turns on the stove). After this, he moves to the kitchen table, he places a dish on the table and eats his meal. Finally, he moves back to the couch to take a rest. The total duration of the trial was about ten minutes.

Figure 3 shows the results of the experiment. The timelines represented are generated by the planner and report both the readings of the sensors (the 3 timelines at the bottom) and the decisions taken by the two components Human and HumanAbstract. At first, the detection of the user on the couch makes the planner to assert the decision **watchingTV** on the component Human (synchronization (1)). In this case, no decision is asserted for the component HumanAbstract, because no synchronization applies. Then the activities **cooking** and **eating** are detected, as the user proceeds to prepare and consume his meal (synchronizations (2) and (3)). As soon as the decision **eating** is asserted on the component Human, synchronization (4) is triggered: the high level **meal** activity is detected. This decision spans from the beginning of the **cooking** activity until the end of **eating**. Finally, when the user goes to rest on the couch, the decision **watchingTV** is asserted once again. This time, since the **meal** activity has been detected, also the higher-level decision **nap** is asserted (synchronization (5)).

Figure 3. Evolutions in time of the components as generated by OMPS during our experiment. The plan monitoring process asserts decisions on components Human and HumanAbstract, while the RFID reader, location and stove sensor components assert decisions that reflect the sensor readings.

The experiment was performed in real time, with re-planning occurring at a frequency of 1 Hz, and with real sensors.

4. Discussion and Conclusion

In this paper, we presented our first results towards the realization of human activity recognition in the PEIS-Home, a service providing household environment. For this purpose, we adapted a constraint-based temporal reasoning framework (OMPS), shifting from the more classic realm of controllable components to encompass sensors and uncontrollable components (e.g., the human user) in the system. We also presented a first implementation of our framework, together with a simple test domain. A real experiment within this domain was detailed to demonstrate the feasibility of our approach.

Our main contribution is the novelty of the approach we take to the problem. We use a temporal planning framework to realize an abductive process which explains sensor data by testing the applicability of temporally-constrained requirements on these data. These requirements are given in the form of OMPS synchronizations, which are grounded on Allen's Interval Algebra. Our approach is complementary to more classical data-driven approaches, as it relies on a domain which models temporal knowledge on human activities. The obvious advantage lies in the fact that domain knowledge does not need be learned, thus allowing us to build modularly and incrementally our synchronizations. Our framework would allow us to both recognize human activities and anomalies in the behaviour of the user, since hazardous situations can be modeled in the same way as normal activities.

It is important to notice that using a knowledge-driven approach is possible if (1) relatively crisp knowledge about how sensor data correlates to human activities is available, and (2) the knowledge representation framework is sufficiently expressive to capture the details and nuances of activity recognition. While point (1) depends on the level of abstraction that is necessary in the particular domain of application, we believe that temporal constraints offer a flexible means to describe how sensor data correlates to human activities, particularly when it is necessary (as in the experiment illustrated in this paper) to model temporal relations between sensor data and human activities across multiple tiers of abstraction.

Our approach was inspired by previous work, such as the RoboCare system [3], in which sensors embedded in a domestic environment are used in conjunction with a schedule execution monitor towards the aim of identifying anomalies in typical user activity patterns. An important difference with the work presented here lies in the fact that RoboCare employs pre-compiled (albeit highly flexible) schedules as models for human behavior. In the present work, we employ a planning process to actually instantiate such candidate schedules on-line starting from general rules.

Our future work will focus on testing our framework with more complex scenarios and to test its capabilities in terms of robustness and scalability to the recognition of multiple activities. We are also extending our approach to allow long periods of continued monitoring, of the order of weeks or months: this requires the ability to forget past data which cannot affect future decisions. Finally, another long term goal would be to adapt our framework to predict, and not only recognize, human activities.

References

[1] J.F. Allen. Towards a general theory of action and time. *Artificial Intelligence*, 23(2):123–154, 1984.

[2] G. Bjorn, H.W. Guesgen, and S. Hubner. Spatiotemporal reasoning for smart homes. In J.C. Augusto and C.D. Nugent, editors, *Designing Smart Homes*, pages 16–34. Springer, 2006.

[3] A. Cesta, G. Cortellessa, M.V. Giuliani, F. Pecora, M. Scopelliti, and L. Tiberio. Caring About the User's View: The Joys and Sorrows of Experiments with People. In *ICAPS Workshop on Moving Planning and Scheduling Systems into the Real World, Providence, RI, USA*, 2007.

[4] A. Cesta and S. Fratini. The Timeline Representation Framework as a Planning and Scheduling Software Development Environment. In *Proc. of 27th Workshop of the UK Planning and Scheduling SIG*, 2008.

[5] A. Cesta, S. Fratini, A. Oddi, and F. Pecora. APSI Case#1: Pre-planning Science Operations in Mars Express. In *Proc. of iSAIRAS-08*, 2008.

[6] R. Dechter, I. Meiri, and J. Pearl. Temporal constraint networks. *Artif. Intell.*, 49(1-3):61–95, 1991.

[7] S. Fratini, F. Pecora, and A. Cesta. Unifying Planning and Scheduling as Timelines in a Component-Based Perspective. *Archives of Control Sciences*, 18(2):231–271, 2008.

[8] A. Goultiaeva and Y. Lespérance. Incremental plan recognition in an agent programming framework. In *Working Notes of the AAAI Workshop on Plan, Activity, and Intention Recognition (PAIR)*, 2007.

[9] V. Jakkula, DJ Cook, and AS Crandall. Temporal pattern discovery for anomaly detection in a smart home. In *Proc. of the 3rd IET Conf. on Intelligent Environments(IE)*, pages 339–345, 2007.

[10] F. Mastrogiovanni, A. Scalmato, A. Sgorbissa, and R. Zaccaria. Assessing temporal relationships between events in smart environments. In *Proc of the Int Conf on Intellingent Environments*, 2009.

[11] R. Muñoz-Salinas, E. Aguirre, and M. García-Silvente. People detection and tracking using stereo vision and color. *Image and Vision Computing*, 25(6):995–1007, 2007.

[12] D.J. Patterson, D. Fox, H. Kautz, and M. Philipose. Fine-grained activity recognition by aggregating abstract object usage. In *Proc. of the 9th IEEE International Symposium on Wearable Computers*, 2005.

[13] M.E. Pollack, L. Brown, D. Colbry, C.E. McCarthy, C. Orosz, B. Peintner, S. Ramakrishnan, and I. Tsamardinos. Autominder: an intelligent cognitive orthotic system for people with memory impairment. *Robotics and Autonomous Systems*, 44(3-4):273–282, 2003.

[14] R.B. Rusu, J. Bandouch, Z.C. Marton, N. Blodow, and M. Beetz". Action recognition in intelligent environments using point cloud features extracted from silhouette sequences. In *Int Symp on Robot-Human Interactive Communication (ROMAN)*, 2008.

[15] A. Saffiotti, M. Broxvall, M. Gritti, K. LeBlanc, R. Lundh, Rashid J., B.S. Seo, and Y.J. Cho. The PEIS-ecology project: vision and results. In *Proc of the IEEE/RSJ Int Conf on Intelligent Robots and Systems (IROS)*, Nice, France, 2008.

[16] M.P. Shanahan. Robotics and the Common Sense Informatic Situation. In *Proceedings ECAI 96*, 1996.

[17] J. Wu, A. Osuntogun, T. Choudhury, M. Philipose, and J. Rehg. A Scalable Approach to Activity Recognition Based on Object Use. In *Proceedings of ICCV 2007. Rio de Janeiro, Brazil*, 2007.