Mining the user profile from a smartphone: a multimodal agent framework

Giuseppe Loseto, Michele Ruta, Floriano Scioscia, Eugenio Di Sciascio, Marina Mongiello
DEI - Politecnico di Bari
via E. Orabona 4, I-70125, Bari, Italy
loseto@deemail.poliba.it, {m.ruta, f.scioscia, disciascio, mongiello}@poliba.it

***Abstract*—Nowadays smartphones play a significant role in gathering relevant data about their owners. Micro-devices embedded in Personal Digital Assistants (PDAs) perform a continuous sensing, the phone call lists, PIM (Personal Information Manager), text messages and so on allow to collect and mine data enough for a high-level description of daily activities of a user. This paper proposes an agent able to perform an automated profile annotation by adopting Semantic Web languages. As a proof of concept, the devised agent has been tested in an Ambient Intelligence (AmI) scenario, *i.e.*, a domotic environment where it interacts with its home counterpart to trigger services best matching the user needs. A toy example is presented as case study aiming to better clarify the proposal while an early experimental evaluation is reported to assess its effectiveness.**

***Keywords*—Ambient Intelligence; Agent-based Data Mining; Semantic Web of Things; Home and Building Automation.**

I. INTRODUCTION

Mobile phones are both pervasive and personal –following the user and having clues about everyday situations– resulting extremely useful to infer a context. Embedded micro-devices (accelerometer, digital compass, gyroscope, GPS, microphone and camera) can be used to extract significant information about the user: GPS location traces, call and SMS lists, PIM (Personal Information Management) records including contacts and calendar, battery charging habits. By leveraging the smartphone processing capabilities, ever-expanding ways to investigate behavioral, spatial and temporal dimensions of the everyday life can be provided. The personal nature of mobile phones suggest they are well suited for pervasive computing, but data they are able to collect and process could be profitably used for a large set of context-aware applications, like the *Ambient Intelligence* (AmI) [1] ones.

This paper presents a smart profiling agent[1] which borrows languages and technologies from the Semantic Web experience to funnel inarticulate raw individual information toward a semantically rich glossary. A crawler agent runs on the user smartphone and performs a multimodal (*i.e.*, involving several heterogeneous data sources) and continuous sensing [2] collecting and processing information without human intervention. The multimodality requires specialized analyses for each kind of collected data. The agent mines the user habits automatically and annotates them in a logic-based formalism to build a daily profile to be further exploited in context-aware knowledge-based applications. The main motivation for adopting an agent-based approach is that the mobile profiler must modulate proactively the amount and complexity of data capture and processing, in order to use energy efficiently. Smart Home and Building Automation (HBA) [3] was selected as proof scenario: the profiling agent sends the inferred preferences to its HBA counterpart so that a logic-based matchmaking session could finalize the adaptation of the environment to user needs.

The remainder of the paper is organized as in what follows. Section II contextualizes the overall multi-agent HBA system motivating the proposed approach before presenting both architecture and algorithms of the profiler agent in Section III. The toy example in Section IV acts as a case study while an early experimental evaluation is reported in Section V. Finally, most relevant related work is discussed in Section VI and concluding remarks and future research are in Section VII.

II. SCENARIO: SEMANTIC-BASED HOME AUTOMATION

The user agent proposed in this paper is intended as a part of a more complex HBA Multi Agent System (MAS) [4] leveraging the semantic-based evolution of the KNX domotic protocol in [5]. It introduced a semantic micro-layer on the top of the stack enabling novel services and functions while keeping a full backward-compatibility with current domestic devices and HBA appliances. The above enhancements allowed to fully describe device features by means of annotations expressed in logic-based languages such as RDF[2] and OWL[3]. The knowledge domain of building automation was conceptualized in a shared ontological vocabulary enabling a rich characterization of home resources and services. The MAS was implemented in Java on a testbed composed of off-the-shelf KNX domotic equipment[4].

The adopted multi-agent system comprised a home mediator agent as well as user and device agents. Each agent adopts the custom service-oriented model sketched in [4, Fig. 4]. Basically, the agent monitors its internal state and inputs; when a significant change occurs, it communicates with the other agents in order to discover suitable services that maximize its utility. The number of both resources/services and agents varied unpredictably (as new users or devices joined or disconnected the system at any time) without redefining the communication paradigm for that.

[1] Project home page: http/sisinflab.poliba.it/swottools/mobile-user-profiler/

[2] RDF (Resource Description Framework) Primer, W3C Recommendation, 10 February 2004, http://www.w3.org/TR/rdf-primer/

[3] OWL 2 Web Ontology Language, W3C Recommendation, 11 December 2012, http://www.w3.org/TR/owl2-overview/

[4] See the related project home page http://sisinflab.poliba.it/swottools/smartbuildingautomation/ for more details.

– The *Mediator Agent* coordinates the explicit characterizations of available services, described w.r.t. a reference ontology modeling the conceptual knowledge for the building automation problem domain. Furthermore, it acts as a broker in order to discover the (set of) elementary services that cover (part of) the request coming from user or device agents.
– The *Device Agents* are thought to run on advanced devices, *i.e.*, home appliances with some computational capabilities and memory availability. Each one can expose one or more semantic descriptions, *i.e.*, functional profiles to be discovered by other agents, or alternatively each of them could issue semantic-based requests to the mediator agent when the device status changes and then require a home reconfiguration.
– *KNX Device Interface Agents* support semantic-based enhancements in case of legacy or elementary appliances, *e.g.*, switches, lamps, and so on. In such cases, there is only a static interaction between agent and device.
– Finally the *User Agents*, running on mobile clients, send requests toward the home environment, in order to satisfy user needs and preferences. W.r.t. the version in [4], an approach for the automated mining of a user profile in charge to that kind of agent is proposed as main contribution of this paper.

III. FRAMEWORK AND APPROACH

Figure 1 sketches the general architecture of the profiling agent. Raw data are extracted from smartphone embedded micro-devices, communication tools and PIM. The data mining life cycle consists of the following subsequent stages: (a) gathering; (b) feature extraction; (c) classification and interpretation; (d) semantic annotation. High-level information about user activities, whereabouts, mental and physical status is inferred and annotated w.r.t. an extension of the HBA ontology in [5]. The mined profile should be finally used to trigger the activation or deactivation of the most appropriate home services. A modular architecture allows to process the various data sources with specialized algorithms. In particular, as shown by icons in Figure 1, three modules fully characterize the agent at the moment: (i) Points of Interest Recognition; (ii) Transportation Mode Recognition; (iii) User Activity Recognition.

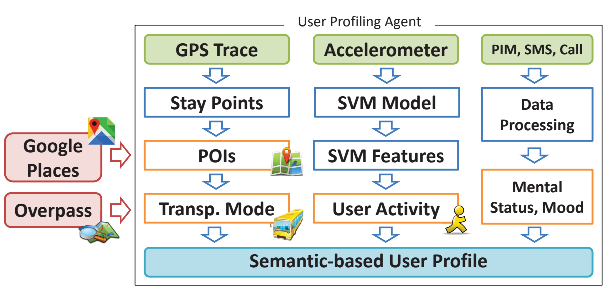
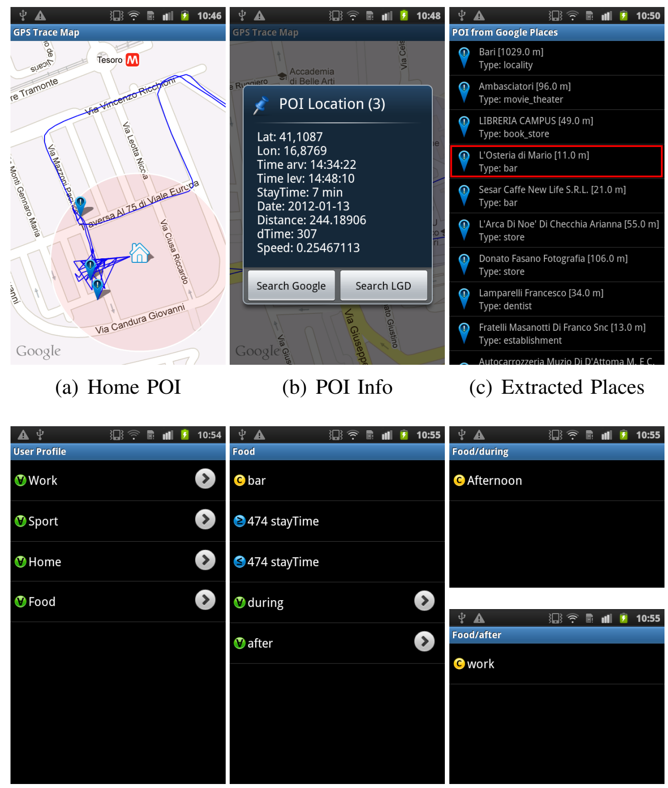


Fig. 1. Reference architecture of the user profiling agent

1. Points of Interest Recognition. A mining algorithm analyzes the smartphone GPS data in order to:
a. identify Stay Points (SPs) through a slightly refined version of the algorithm in [6];
b. for each SP, retrieve the nearest Point Of Interest (POI) via reverse geocoding queries to *Google Places*[5] Web service;
c. associate a "place category" to each POI, so as to further infer the kind of user activity;
d. enrich the daily user profile conjoining all detected activities, described w.r.t. a proper HBA ontology.
A SP represents a narrow geographic region where a user stands for a while. In particular, given two subsequent detected GPS locations P_1 and P_2, a SP satisfies both the following constraints: (i) maximum distance $d(P_1, P_2) < D_{max}$; (ii) minimum time difference $|T_1 - T_2| > T_{min}$, where the thresholds were set to $D_{max} = 200m, T_{min} = 350s$. An empirical evaluation was executed to assign the thresholds values granting the highest precision of the SP recognition algorithm.



(a) Home POI (b) POI Info (c) Extracted Places

(d) Profile mining (e) *Food* place detail (f) Daily stay period and location visited before

Fig. 2. Screenshots of the GPS profiler

Figure 2 shows the GUI of the profiler prototype on the GPS-side. The daily GPS trace is drawn on Google Maps together with detected SPs, depicted as markers on the map in Figure 2(a). The *Home* and *Workplace* POIs are set by the user in a preliminary configuration step. As said, the SP classification leverages a Web-based reverse geocoding service: after comparing Google Places and *LinkedGeoData* (LGD) [7] (see Section V for further details) the first one service has been chosen at the moment, since it provides more available POIs even if LGD often seems to be more accurate. In the example reported in Figure 2(c), the agent selected a SP near to the *Politecnico di Bari* and all the nearby POIs were retrieved by means of the Google Places API. The main category of the nearest POI is used as label of the retrieved location. Starting from the Google Places classification[6], the

[5] http://developers.google.com/places/

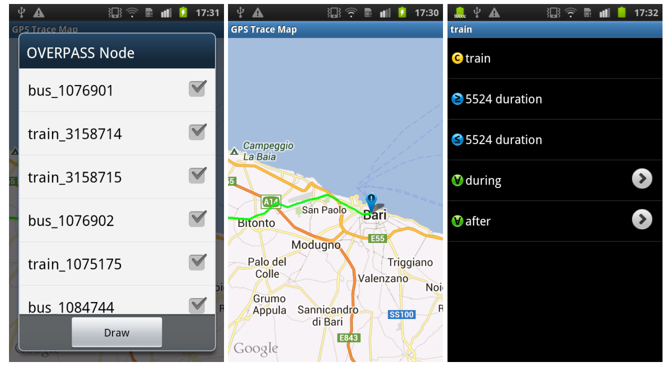
[6] http://developers.google.com/places/documentation/supported_types/

reference ontology for domotics in [5] has been extended to include a places taxonomy. Finally, as reported by the Figure 2(d), a profile is generated through the conjunction of location information. As shown in Figure 2(e), each SP description contains an ontology class related to the specific location the user visited, the overall time spent there (in seconds), the daily period and the place visited before, if present (Figure 2(f)).

2. Transportation Mode Recognition. GPS data are exploited also to detect the transportation mode adopted by the user when moving during a day. Four transportation modes are supported: bus, train, car or walking. A pre-processing splits the whole daily GPS trace $\mathcal{P} = \{T_1, \ldots, T_n\}$ in trajectories $\mathcal{T}_i$. In turn, each trajectory $\mathcal{T}_i = \mathcal{Q}\{POI_i, POI_{(i+1)}\}$ consists of a set of GPS points $\mathcal{Q}$ included between two subsequent POIs. Starting from the trajectories set, the transportation mode detection is based on two reference parameters: (i) the *walking speed* threshold (WS_{th}), set to an average value of 2 m/s (*i.e.*, 7.2 km/h); (ii) the *minimum correspondence ratio* (CR_{min}) between user trajectories and bus/train routes, set to 0.8 (*i.e.*, at least a 80% correspondence is required). Also in this case, an experimental evaluation was performed to select the most suitable threshold values. The algorithm for detection progresses along the following stages:
a. For each trajectory T_i, the average user speed is evaluated. If it is lower than WS_{th} then walking mode is detected.
b. Otherwise, the algorithm queries OpenStreetMap[7] (OSM) via the Overpass API[8] to retrieve all available bus and train routes ($Rs = R_{bus} \cup R_{train}$) in a *bounding box* covering the geographical coordinates of the GPS points in T_i. Figure 3(a) shows an example for that.
c. A comparison between the GPS points of the user trajectory and the retrieved routes is performed. In case of a correspondence ratio greater than CR_{min} with a bus or train path, the trajectory T_i is associated to a bus or train mode, respectively (Figure 3(b)).
d. Finally, if the detected mean is neither walking nor train nor bus, then the car mode is selected.
Each transportation mode is associated to a semantic-based annotation fragment which includes a given class of the ontology, further extended to include also concepts and properties about user movements. Moreover, the description will include the overall time –in seconds– the user spent during the day for moving, the daily period and possible means of transport used before. Figure 3(c) shows the details about the user profile section related to a transfer by train.

3. User Activity Recognition. Beyond the above components, the profiling agent is completed by a module to detect some user activities. In particular, at the moment the following elementary actions can be discovered: sitting, standing, walking, walking upstairs and dowstairs. Starting from data acquired from the smartphone accelerometer and gyroscope, a supervised Machine Learning (ML) approach is adopted, exploiting the Support Vector Machines (SVM) classifier in [8]. W.r.t. the original approach, the classifier was simplified to improve its efficiency on PDAs and to reduce the training time. The early 568 features used on the dataset[9] associated to [8] as input

[7]http://www.openstreetmap.org/
[8]http://wiki.openstreetmap.org/wiki/Overpass_API
[9]http://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones



(a) Overpass routes (b) Train Mode (c) Train Mode details

Fig. 3. Screenshots of the Transportation Mode profiler

#	Feature description
1	tBodyAcc correlation(X,Y)
2	tGravityAcc mean(X)
3	tGravityAcc mean(Y)
4	tGravityAcc max(Z)
5	tGravityAcc min(X)
6	tGravityAcc energy(X)
7	tBodyGyro iqr(Z)
8	tBodyGyroJerk entropy(X)
9	tBodyGyroJerk entropy(Z)
10	tBodyAccJerkMag iqr(X,Y,Z)
11	tBodyGyroJerkMag energy(X,Y,Z)
12	fBodyGyro max(Y)
13	fBodyGyro max(Z)
14	fBodyGyro skewness(Z)
15	fBodyAccMag std(X,Y,Z)
16	fBodyAccMag energy(X,Y,Z)

t=time domain, f=frequency domain, Jerk=derived in time, Mag=Euclidean norm, iqr=Interquartile range

TABLE I. FEATURES SUBSET FOR THE SVM CLASSIFIER

for the classifier were reduced to 16 (see Table I) by applying the Recursive Feature Elimination (RFE) algorithm proposed in [9].

A training set composed by sensor raw data has been used to let the classifier learn directly on the mobile device. The smartphone used for the experimental evaluation is equipped with an accelerometer and a gyroscope measuring both the 3-axial linear acceleration and the angular velocity (*tAcc-XYZ* and *tGyro-XYZ*, respectively) at a fixed sampling rate of 25 ms, which is adequate to identify a human body motion. The collected data are subsequently processed through two first-order low-pass filters. The first one is used to reduce noise, while the second filter splits the acceleration signal into body and gravity components (*tBody* and *tGravity*). The classifier has been implemented using *Weka-for-Android*[10], an Android port of Weka [10]. The training set has been built fastening the smartphone in vertical position as reference; after the SVM training, the recognition process starts. Data are sampled in fixed-width sliding windows of 2.5 s (*i.e.*, 100 samples) with 50% overlap, and processed as described above. From each window, a vector with the 16 features in Table I is obtained by computing the extracted accelerometer and gyroscope data in the time and frequency domain. Finally, an energy saving strategy is implemented to avoid unnecessary data capture: after each activity recognition AR_i, a pause WP_i is waited

[10]https://github.com/rjmarsan/Weka-for-Android

for. WP_i is defined as:

$$WP_i = \begin{cases} 0sec & \text{if } AR_i \neq AR_{i-1} \\ 2.5sec & \text{if } AR_i = AR_{i-1} \\ (WP_{i-1} * 2)sec & \text{if } AR_i = AR_{i-1} = AR_{i-2} \end{cases}$$

In this way, if the classifier consecutively detects two similar activities, then the data sampling is stopped for 2.5 seconds. This value is doubled in case of additional similar recognitions, up to a maximum value of $WP_i = 80s$. Otherwise, the waiting period is reset to zero when a different action is detected. The rationale is that users usually perform similar activities in a short period –consider for example the case of sitting and walking– so a continuous data gathering could be often avoided.

The vector containing the extracted features is then used as input of the trained SVM model. Finally, the user profile is enriched with the annotations related to the detected activities. For each of them it will be also considered the overall stay time and the daily period.

IV. Case study

In order to clarify the rationale behind the proposed approach and to let emerge the goal of the profiling agent, the following daily scenario is considered as example. *The user leaves home early in the morning to go to work. He remains at office until lunch, then reaches a bar for a fast meal. Afterward, he comes back to work, then goes to the gym in the evening and finally returns home late at night*. The profiling agent extracts the daily location sequence reported in Table II. Particularly, Home and Office POIs are mapped to the user profile directly as *Home* and *Work* activities; Bar is identified as a *Food* place; Gym is associated to the *Sport* place category. The agent also recognizes the adopted means of transport and the duration of each trajectory.

Route	Type	Duration (min)
Home → Office	car	30
Office → Bar	walk	4
Bar → Office	walk	5
Office → Gym	car	11
Gym → Home	car	21

TABLE II. Daily user locations and routes

Along the day, the agent also detects the activities of the user: he was seated for about 6 hours (*e.g.*, at work, within the car, during lunch), walked for 35 minutes (*e.g.*, to reach the bar or for short strolls) and was standing for 15 minutes. As a result of the mining and annotation processes, the following profile is extracted (expressed in Description Logic [11] notation w.r.t. the reference ontology)[11]:

User_Daily_Profile $\equiv \forall\ wasAtHome.HomeActivity \sqcap \forall\ wasAtWork.WorkActivity \sqcap \forall\ wasInFoodPlace.FoodActivity \sqcap \forall\ wasInSportPlace.SportActivity \sqcap \forall\ movedByCar.CarMode \sqcap \forall\ movedByWalk.WalkMode \sqcap \forall\ wasSitting.SittingActivity \sqcap \forall\ wasWalking.WalkingActivity \sqcap \forall\ wasStanding.StandingActivity$

HomeActivity $\equiv Home \sqcap \forall\ during.(Morning \sqcap Night) \sqcap \forall\ after.Gym \sqcap =_{1945}\ stayTime$

WorkActivity $\equiv Work \sqcap \forall\ during.(Morning \sqcap Afternoon) \sqcap \forall\ after.(Home \sqcap Bar) \sqcap =_{32470}\ stayTime$

[11]Due to space constraints, some sections have been voluntarily omitted.

FoodActivity $\equiv Bar \sqcap \forall\ during.Afternoon \sqcap \forall\ after.Work \sqcap =_{474}\ stayTime$

SportActivity $\equiv Gym \sqcap \forall\ during.Evening \sqcap \forall\ after.Work \sqcap =_{5362}\ stayTime$

WalkMode $\equiv Walk \sqcap =_{2115}\ moveTime \sqcap \forall\ during.Afternoon \sqcap \forall\ after.Car$

SittingActivity $\equiv Sitting \sqcap =_{21436}\ moveTime \sqcap \forall\ during.(Morning \sqcap Afternoon \sqcap Evening)$

The above generated profile will be adopted by the user agent to negotiate with the mediator agent at home the environmental situation best fitting needs and mood of the inhabitant via a semantic-based matchmaking. The elementary services and appliances covering the mined user profile as much as possible are automatically activated (or in case deactivated) to increase the overall MAS utility. As an example of this phase, let us consider the following available home services/resources:

CookingService $\equiv Service \sqcap \forall\ wasInSportPlace.(\ >=_{1800}\ stayTime) \sqcap \forall\ wasAtHome.(\ \forall\ after.(Sport \sqcap \neg Food)) \sqcap \forall\ suggestedForFeeling.Hungry$

SoftLightLevel $\equiv LightLevelRegulation \sqcap \forall\ wasAtWork.(\ >=_{10800}\ stayTime) \sqcap \forall\ wasAtHome.(\ \forall\ after.\ \neg Relax) \sqcap \forall\ suggestedForStamina.MentallyTired \sqcap \forall\ suggestedForDisease.Headache$

PlayMusic $\equiv Service \sqcap \forall\ wasAtHome.(\ \forall\ after.(\ \neg Work \sqcap Relax) \sqcap \forall\ during.\ \neg Night) \sqcap \forall\ suggestedForStamina.Rested \sqcap \forall\ suggestedForDisease.\ \neg Headache$

It should be noticed that service annotations are described in terms of both user features (such as a physical status, mood and health) and daily events which cause the activation. In this way, a service/resource selection can be performed through the matchmaking against the user profile. For example, a cooking service is activated not only if the user explicitly declares he is hungry, but also if the user agent detects he comes back home after a sport activity, performed for more than 30 minutes (expressed in seconds), without eating anything before. In a similar way, a soft lighting setting is selected to improve the comfort at home in case the user is mentally tired and he spent more than 3 hours at work not followed by a restful activity. The extracted user profile can also lead to a deactivation of previously enabled services. For example, the music service is normally activated to welcome the owner at home, but it is unsuitable if the user comes back during the night and in that case it must be turned off.

The above case study is purposely simplified in order to make the presentation of the proposed approach clear and short. In real scenarios, more articulated user profiles and service descriptions can be used.

V. Experiments

An overall evaluation of the proposed approach has been carried out following a reference user for a period of 14 months. Results reported here refer to the first 60 days of observation. In particular, only the days –24 in the evaluated dataset excerpt– with at least one Stay Point different from Home or Workplace have been selected for further investigation. The profiling agent has been tested on a smartphone equipped with an ARM Cortex A8 CPU at 1 GHz, 512 MB RAM, a 8 GB internal storage memory, and Android 2.3.3

as operating system. Done experiments basically aimed to measure: (i) the amount of data retrieved from services on the Web; (ii) the turnaround time (for which each test was repeated four times taking the average of the last three runs); (iii) the memory usage (for which the final result was the average of three runs). This experimental analysis only focuses on the user profiling aspects: [4] reports on evaluation of the remaining elements of the reference HBA MAS.

Figure 4 shows the total number of stay points detected with the mining algorithm compared with the overall GPS coordinates composing a daily trace. It can be noticed that the user agent collects 53 GPS points per day on average, detecting about 3 relevant SPs.

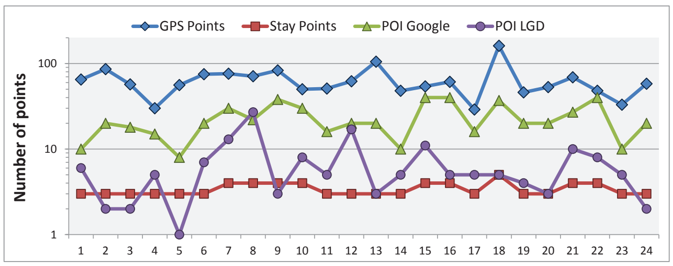


Fig. 4. GPS points, detected SPs and retrieved POIs

Starting from detected SPs, the results of Google Places and LGD services have been compared in terms of number of retrieved POIs in the neighborhood of each SP. As shown in Figure 4, Google Places usually returns 16 POIs w.r.t. 5 POIs on average retrieved by LGD, so an accurate identification of the locations the user visited is more likely. Nevertheless, as reported in Figure 5, in some cases the LGD replies are longer even though it returns fewer POIs. This is due to the LGD response format including, for each point, information annotated according to *Linked Data* principles [12]: Google Places uses 830 B per POI on average, whereas LGD uses 1.56 kB.

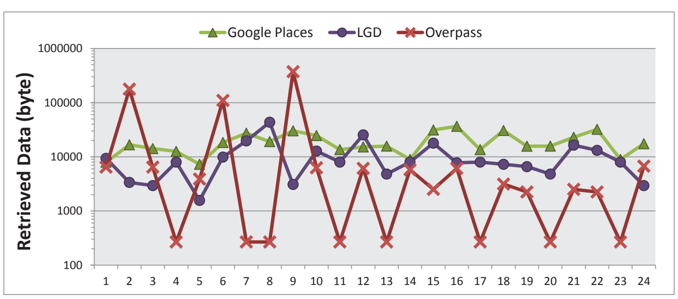


Fig. 5. Retrieved Data

The time required by the main processing steps for POIs recognition (GPS traces parsing; SPs detection; Google Places/LGD services querying; profile enrichment), transportation mode detection (Overpass service querying; traces comparison; profile enrichment) and activity recognition are reported in Figure 6. Google Places is slightly slower than LGD, but this is due to the greater amount of retrieved POIs. Considering Google Places as reference service, the agent spends about 1.2 s to retrieve the POIs from a detected SP.

Activity	A	B	C	D	E	Recall %
A Sitting	**340**	0	0	0	0	100
B Standing	0	**98**	0	1	0	98.9
C Walking	1	0	**70**	0	3	94.6
D Walking Upstairs	0	0	2	**125**	5	94.7
E Walking Downstairs	0	0	0	4	**130**	97.0
Precision %	99.7	100	97.2	96.2	94.2	**98.0**

TABLE III. CONFUSION MATRIX

In particular, the last step took about 1.15 s (49% of total time) to parse the ontology and create the semantic-based annotation. The remaining steps require only the 3% of the overall turnaround time, as these procedures use elementary data structures stored in the device main memory. For the transportation mode detection, only 1.7 s were spent to query the Overpass service, while traces comparison is one of the slower operations, needing 3.4 s. The activity recognition process has a very short turnaround time. After a preliminary task (required to train the SVM classifier) taking about 5.6 s and performed when the profiling agent starts, this module needs only 45 ms to extract the 16 reference features for each windows and 6 ms to detect the user activity. Finally, a daily profile was completely composed in about 1.2 seconds.

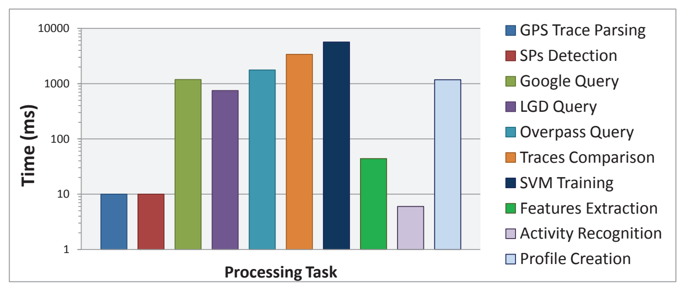


Fig. 6. Processing Time

A further evaluation of the activity recognition module required to measure precision and recall of the classifier. 100 datasets of activities containing a similar number of samples per class have been used. The confusion matrix shown in Table III reports on the weighted precision of the classifier and on single precision and recall values for each activity. It is referred to a single specific dataset with 779 sample vectors. However all confusion matrices for different tests showed similar outputs, varying slightly in the classification results. It is possible to notice that the classifier precision and recall are very high despite the usage of a small set of features.

RAM usage trend was also evaluated and results are shown in Figure 7, where memory peaks are reported. The profiler agent needs very low memory, only 4.2 MB on average, a satisfactory value for current mobile devices.

VI. RELATED WORK

The recent popularization of smartphones equipped with a wide range of embedded sensors and adequate processing capabilities has attracted increasing research efforts toward mobile sensing. Lane *et al.* [2] proposed a survey on existing algorithms, applications, and systems. In addition, many pervasive frameworks were defined to collect and capture the user's

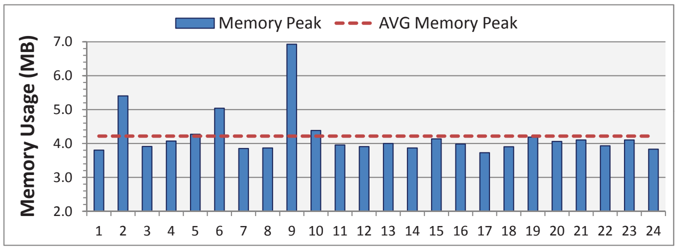


Fig. 7. Main memory usage trend

context via cellphones in latest years: remarkable works are *ContextPhone* [13], *UbiqLog* [14] and *LifeMap* [15]. The agent proposed here aims to improve upon these works by leveraging the multimodality aspect: the implemented prototype retrieve information from a data source richer than the above systems, even though further mining modules have been planned but not integrated yet. A comparison should be carried out also with respect to commercial location and context-aware mobile software: trekking and fitness applications like *Google MyTracks*[12] and *Endomondo Sportstracker*[13]; personalized assistants like *Google Now*[14] and *Xme*[15]. Nevertheless, these tools either require explicit user interaction or define context just by means of GPS location and time of day, hence they are quite far off the agent proposed here which uses more parameters and automatically recognizes a larger variety of contexts.

The activity recognition from accelerometer by means of machine learning is a frequent sensing application. Among other proposal, noteworthy are [16], [8] where smartphone accelerometer data are used to classify six common activities. With reference to context extraction via GPS data analysis, there are many approaches in literature. For example Zheng *et al.* [17] model multiple individuals GPS trajectories with a tree-based hierarchical graph to mine location history and travel sequences in a given geospatial region. In [6] mobile phones are used as sensors to collect location information. Places are first grouped using a time-based clustering technique to discover *stay points*; then the stay points are clustered in *stay regions* through a grid-based algorithm. In [18] a large-scale dataset is collected from 114 users over 18 months.

In the above cited works, however, the knowledge gap between acquired data and the understanding of human behavior is still huge. Stay points and movement patterns require to be interpreted to extract a user profile, implicitly providing knowledge about the user habits. Noteworthy attempts to enrich movement trajectories with semantics are in [19] and [20]. An ontology-based approach for a semantic modeling of trajectories is also proposed in [21]. Trajectories are seen as composed by three main elements: stops, moves and begin-ends. Each part is described through an annotation referred to a domain ontology and time information are also exploited to annotate activities to enable rule-based queries and to help users validate and discover moving objects.

Although previous solutions add a machine-understandable meaning to data collected by smartphones, a subsequent exploitation in an articulated AmI framework is still missing. Usually, collected data are only used to indicate detected user conditions or activities through messages or alerts displayed on the mobile phone. On the contrary, in the approach proposed here, the ontology-based characterization of user activities is used as an input for a context-aware HBA MAS [4], enabling a direct environment adaptation and a negotiation between user and home agents. This feature is not possible for any other current user profiler.

[12]http://www.google.com/mobile/mytracks/
[13]http://www.endomondo.com
[14]http://www.google.com/landing/now/
[15]http://xndme.com/

VII. Conclusion and Future Work

The paper presented a lightweight agent able to mine data collected by embedded micro-devices, logs and applications of a smartphone to build a semantic-based daily profile of its user. According to the AmI paradigm, such a description can be exploited to transparently adapt the environment to user preferences, implicitly inferred. In the matter in question, the agent interacts in a multi-agent framework for Home and Building Automation, grounded on knowledge representation theory and reasoning technologies. It has been designed and then implemented as an Android application and experiments in a concrete case study proved its feasibility and effectiveness.

Future work will include a more extensive experimental campaign involving several different users to be profiled and new performance indicators. Particularly, both battery drain and storage peaks will be taken into account to assess the feasibility of a continuous data collection and mining and to compare the provided framework with existing approaches. Also the exploitation of an agent-based framework w.r.t. to classical approaches will be posed under investigation to verify if it results in a more accurate profiling action. Finally, future research will be also devoted to the integration of the current multimodal information. A fusion of information coming from data sources which now are distinct and independent will be pursued in order to reach a more accurate and precise user characterization.

Acknowledgment

The authors acknowledge partial support of Italian PON project *Res Novae* and EU PO Apulia region FESR project *UbiCare*.

References

[1] D. J. Cook, J. C. Augusto, and V. R. Jakkula, "Ambient intelligence: Technologies, applications, and opportunities," *Pervasive and Mobile Computing*, vol. 5, no. 4, pp. 277 – 298, 2009.

[2] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell, "A survey of mobile phone sensing," *IEEE Communications Magazine*, vol. 48, no. 9, pp. 140–150, Sep. 2010.

[3] G. Loseto, F. Scioscia, M. Ruta, and E. Di Sciascio, "Semantic-based Smart Homes: a Multi-Agent Approach," in *13th Workshop on Objects and Agents (WOA 2012)*, ser. CEUR Workshop Proceedings, F. De Paoli and G. Vizzari, Eds., vol. 892, Sep 2012, pp. 49–55.

[4] M. Ruta, F. Scioscia, G. Loseto, and E. Di Sciascio, "Semantic-based resource discovery and orchestration in home and building automation: a multi-agent approach," *IEEE Transactions on Industrial Informatics*, 2013, to appear.

[5] M. Ruta, F. Scioscia, E. Di Sciascio, and G. Loseto, "Semantic-based Enhancement of ISO/IEC 14543-3 EIB/KNX Standard for Building Automation," *IEEE Transactions on Industrial Informatics*, vol. 7, no. 4, pp. 731–739, 2011.

[6] R. Montoliu, J. Blom, and D. Gatica-Perez, “Discovering places of interest in everyday life from smartphone data,” *Multimedia Tools and Applications*, pp. 1–29, 2012.

[7] C. Stadler, J. Lehmann, K. Höffner, and S. Auer, “LinkedGeoData: A Core for a Web of Spatial Open Data,” *Semantic Web Journal*, vol. 3, no. 4, pp. 333–354, 2012.

[8] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra and Jorge L. Reyes-Ortiz, “Human Activity Recognition on Smartphones using a Multiclass Hardware-Friendly Support Vector Machine.” in *Workshop of Ambient Assisted Living (IWAAL 2012)*, 2012.

[9] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, “Gene selection for cancer classification using support vector machines,” *Machine Learning*, vol. 46, pp. 389–422, 2002.

[10] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, “The WEKA data mining software: an update,” *SIGKDD Explor. Newsl.*, vol. 11, no. 1, pp. 10–18, 2009.

[11] F. Baader, D. Calvanese, D. Mc Guinness, D. Nardi, and P. Patel-Schneider, *The Description Logic Handbook*. Cambridge University Press, 2002.

[12] C. Bizer, T. Heath, and T. Berners-Lee, “Linked Data - The Story So Far,” *International Journal on Semantic Web and Information Systems*, vol. 5, no. 3, pp. 1–22, 2009.

[13] M. Raento, A. Oulasvirta, R. Petit, and H. Toivonen, “Contextphone: A prototyping platform for context-aware mobile applications,” *IEEE Pervasive Computing*, vol. 4, no. 2, pp. 51–59, Apr. 2005.

[14] R. Rawassizadeh, M. Tomitsch, K. Wac, and A. Tjoa, “Ubiqlog: a generic mobile phone-based life-log framework,” *Personal and Ubiquitous Computing*, pp. 1–17, 2012.

[15] J. Chon and H. Cha, “LifeMap: A Smartphone-Based Context Provider for Location-Based Services,” *IEEE Pervasive Computing*, vol. 10, no. 2, pp. 58–67, Apr. 2011.

[16] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, “Activity recognition using cell phone accelerometers,” *ACM SIGKDD Explorations Newsletter*, vol. 12, no. 2, pp. 74–82, 2011.

[17] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma, “Mining Interesting Locations and Travel Sequences From GPS Trajectories,” in *Proceedings of the 18th International Conference on World Wide Web*, ser. WWW ’09. New York, NY, USA: ACM, 2009, pp. 791–800.

[18] T. M. T. Do and D. Gatica-Perez, “The Places of Our Lives: Visiting Patterns and Automatic Labeling from Longitudinal Smartphone Data,” *IEEE Transactions on Mobile Computing*, 2013, PrePrints.

[19] C. Renso, M. Baglioni, J. Macedo, R. Trasarti, and M. Wachowicz, “How you move reveals who you are: understanding human behavior by analyzing trajectory data,” *Knowledge and Information Systems*, pp. 1–32, 2012.

[20] C. Parent, S. Spaccapietra, C. Renso, G. Andrienko, N. Andrienko, V. Bogorny, M. L. Damiani, A. Gkoulalas-divanis, J. Macedo, N. Pelekis, Y. Theodoridis, and Z. Yan, “Semantic Trajectories Modeling and Analysis,” *ACM Computing Surveys*, vol. 45, no. 4, 2013.

[21] R. Wannous, J. Malki, A. Bouju, and C. Vincent, “Time Integration in Semantic Trajectories Using an Ontological Modelling Approach,” in *New Trends in Databases and Information Systems*, ser. Advances in Intelligent Systems and Computing, M. Pechenizkiy and M. Wojciechowski, Eds. Springer Berlin Heidelberg, 2013, vol. 185, pp. 187–198.