User Engagement Engine for Smart City Strategies

C. Badii, P.Bellini, D.Cenni, A. Difino, P. Nesi, M. Paolucci

University of Florence, Department of Information Engineering,
DISIT Lab, http://www.disit.org, http://www.sii-mobility.org, <name.surname>@unifi.it

Abstract— **The new challenges in the smart city context are mainly related to the stimulation of the city users towards taking more sustainable behaviors, in mobility and energy. The state of the art in this case is mainly focused on classical smart city solution for informing the city users and or for engaging them with specific wired rules toward virtuous models. And not using flexible languages and predictive models, pushing them towards a larger range of virtuous habits. On this regards, the main problems are the computation of user behavior via data analytic (semantic computing, machine learning), as well as the formalization of strategies via simple and well formalized language for producing engagements to the city users, which can be understood by city operators. In this paper, a solution for city users engagement is studied and implemented for Sii-Mobility Smart city national project in Italy has been presented. The solution has been implemented thanks to the exploitation of Km4City model and semantic computing. The paper also presents the validation of results about the effective usage of the solution by providing some statistical evidence about the efficient assessment of user behavior and of engagement rules acceptance rate.**

Keywords— smart city, user behavior, user engament language, sustainable smart city, smart city strategies

I. INTRODUCTION

Most of the smart city solutions are focused on providing services and information to the city users about transport, energy, tourism, culture, etc. The simpler and faster approach for the entry level smart city implementation could be to expose open data portal -- e.g., [1] -- with the hope of seeing someone producing services exploiting them. The further step consists in providing smart city Application Programming Interface (API) to access at single data set or data aggregations. For example: CitySDK makes transformations on each dataset in order to obtain and manage uniform data [2]; Transport.API provides aggregated open data in the UK about transport information [3]; Navitia.io is an open source project exploiting OpenDataSoft for open data aggregation in France [4]; Km4City Smart City API provides semantically integrated information for app and services [5] with its API [6]. The successive step in Smart City implementation consists in deeply understand the city usage and thus the user behavior [7], [8]. The analysis of the city users' behavior, on the basis of the knowledge of the city infrastructure and context should allow identifying possible directions for the city improvements. Once the objective of city improvement is identified, a suitable strategy has to be put in place and persecuted to change city user attitudes. Thus, strategies for city improvement may be defined to a more sustainable exploitation of city services by the city users. For example, strategies to: reduce the number vehicles in certain areas, increase the usage of public transport

in certain time slots, move tourists in diversifying paths in the city visit, go also on minor museums, consume less energy in certain time slots, select less busy parking areas, etc.

A. Related work

On this regard, the first step consists in being capable to identify the right users that should change their habits in using the city, reach them with convincing stimuli that may change their habits (especially for citizens and commuters and less relevant for other city user kinds). Well known solutions and techniques produce suggestions and recommendations guessing "what the user would like", by similarity [9], by indexing [5], by filtering, etc., may be according to the advertiser strategies The engagement should be more capable to understand the story of the user behavior, and predicting possible new future attitudes [10] according to the city strategies. The context of smart city is much more complex than the simple media advertising since spatial reasoning is added to the classical temporal reasoning [8]. Sensors and mobile phones are capable to monitor the context in which the city users move around and substantially can instrument city users for engagement [11]. Level of engagement, point of engagement, disengagement and reengagement are some of the phases in which the user is involved during the day. In literature, several methods analyze and evaluate users' participation, measure the focus attention, the endurability and the novelty [12] as well as retention rate, click depth, popularity [13] has been deeply studied to evaluate how good an application is on taking care of the user's experience; and for gamification technics for a more enjoyable and smooth experience [14].

B. Paper organization

In this paper, an integrated solution to pass from strategies to stimuli at the city users via mobile phone is presented. The solution is grounded on assessing user behavior, deriving contextual information, computing conditions of users, detecting city users changes, providing engagements, collecting feedbacks and follow ups to perform analysis. The whole approach has been developed and enforced into Sii-Mobility Smarty City National project on mobility and transport of the MIUR (Italian Ministry of University and Research) (http://www.sii-mobility.org), with its mobile App and smart city infrastructure based on the Km4City ontological model (http://www.km4city.org) [5] and Smart City API [6].

The paper is organized as follows. In section 2, main requirements and architecture are reported. Section 3 presents the User Engagement Engine providing details about conditions, user behavior data analytics, actions, and feedback rules. In Section 4, implementation notes are reported. Section 5 reports the experimental results putting in evidence the

effectiveness of the solution in understanding the user behavior, with trajectories and precision in providing specific engagement rules to city users. Conclusions are drawn in Section 6.

II. REQUIREMENTS AND ARCHITECTURE

Most of the smart city solutions for sustainable smart city are focused on implementing specific strategies and cases: for sustainable mobility [15], energy, etc. To enable the definition and the implementation of strategies we propose an engagement engine to keep track of the user evolution and context, providing mechanism of reward and gamification (flow experience) [16], promoting virtuous activities and discouraging other ones. The aim of the Sii-Mobility User Engagement Engine (UEEngine) is to cover a range of cases by proposing a flexible language to formalize strategies and to pass from them to actual implementation of mobile platform to stimulate changes into city users. The main requirements of an engagement solution for sustainable mobility would be as follow. Thus, an UEEngine should be capable to:

- **assess the user behavior** according to its status and evolution in space and time: personal points of interests, trips, modality of moving, etc.

- **identify the city users'** habits which may match with the behavior to be corrected according to the city strategies. Collective city users' habits are the recurrent activities that, if not sustainable, may be worth to change thus creating a relevant and tangible benefit for the society.

- **provide a sort of programming visual language** to formalize the strategies to stimulate the city users in changing habits. The formal language has to be: (i) easy understandable by city operators, (ii) expressive and flexible to allow the formalization of a wide range of cases and actions such as sending: information, guidance/ help, requests, bonus, stimuli, soundages, etc., and to track the evolution of engagement discourse established with the city users.

- **provide tools for assessing** the city conditions before and after the engagement campaign. This would allow following up the effects of the engagement campaign, and related investments.

The main architecture of the UEEngine is highlighted in Figure 1. From the left side, the City Operator may define strategies by using an Engagement Rule Editor in which also define eventual bonus, tickets to be spent on strategies according to some relevant evidence (for example, a pay per: click, attitude chance, wrong attitude detection, etc.). Their concerns are to reach some goals that can be economical or sociological, for sure not technological at all. Some of the Strategies can reward the city users when they change behavior, or just stimulate with bonus to change the habits (e.g., we offer you a free ticket for the parking area if you park outside instead of going downtown). To support the definition of the action rules, the UEEngine rely on a digital Wallet, a software realized internally ad-hoc for a generic engagement purpose, to distribute digital points, upon observation on user

activity, and to distribute goods or prize in exchange of the collected digital points.
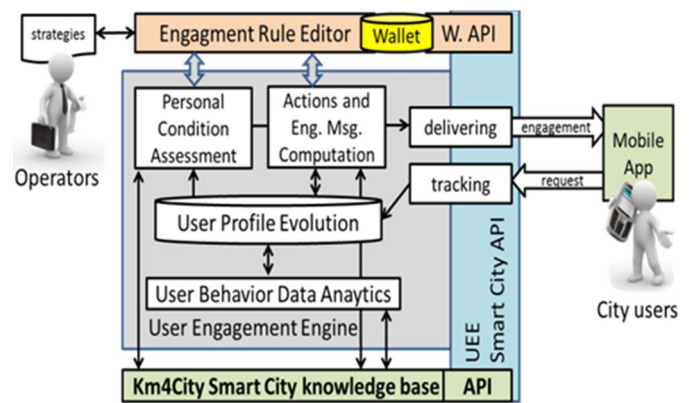

Figure 1 - Sii-Mobility User Engagement Architecture

The defined Strategies are automatically translated in Conditions to be assessed/detected and Actions propagated to the final City Users based on User Profile Evolution directly observed from the Mobile App, from User Behavior Data Analytics and from the city context which can be directly accessible on Km4City [5]. The engagement rules are defined using the Drools Workbench, the web User Interface of the Drools tools-suite.

Therefore, data about changes in the City Users status are collected continuously, and the system receives requests from the Mobile Application periodically. In some cases, the UEEngine may have elaborated some action according to the detected changes, and thus may be ready to provide short or long term engagements. The list of the engagements sent to the users are recorded for future analysis, avoid replications, and matching the follow up, etc. If the user really read/consume an engagement, it will be labelled as Viewed. Thus, the User Behavior Data Analytics may assess the follow up, as well as compute eventual bonus, etc. The City User may be interested to verify the status of his/her Wallet via specific API.

The UEEngine keeps an updated version of the user's context and, on top of this, monitors which conditions (specified in the engagement rules' set) are valid. Whenever all the conditions specified in an engagement rule became true, the platform prepare and delivery the specified action (provide an engagement to the user). At the same time, the platform monitor if/when the delivered engagement is really carried out (i.e. if a request of survey is sent to the user, the platform checks if the survey has been viewed, filled and submitted) and eventually update the context according the information specified in the feedback rules' set.

III. USER ENGAGEMENT ENGINE

UEEngine takes in charge the strategies which are translated into two kinds of Rules:

- Engagement Rules: the combinations of the conditions with the actions;

- Feedback Rules: the kind of User Profile Evolution update that may occurs in case of engagement/ disengagement directly computed by the User Behavior Data Analytic.

A Strategy is converted in one or more Engagement Rules [17] in the form of:

Rule: IF <set-of-firing-events> AND <set-of-conditions> THEN <action>

Where: each <firing event> (at least one) is a special formula $F(C^{ts,te}(u))$ that checks (on User Profile Evolution database) if some context's characteristics of user u assume a certain value in a specific time range [ts,te]. They are the firing conditions to evaluate the <set-of-conditions> (Personal Condition Assessment) part of the rule – e.g., the arrival in a place, the answer to a question. The firing conditions have to be verified for each user with a sufficient frequency in order to be ready to produce the actions, stimulating the city user in the right moment, not too far from the occurrence of the firing condition. The actions may be more or less convincing suggestions and in some case may provide a value, such as a bonus, a discounts, an event tickets, etc.

In the following subsections, Conditions, User Behavior Data Analytics, Actions, and Feedback Rules are better described and formalized. Please note that Actions issue Engagement Messages to users.

A. Personal condition assessment

A Condition is the conjunction of a set of contextual characteristics $C_1(u, t), C_2(u, t), ... C_n(u, t)$, which represent the characteristics 1,2,..n of the User Profile Evolution in which each user u may be at a certain time instant, t:

$$C(u, t) = \{C_1(u, t), C_2(u, t), … C_n(u, t)\}$$

The set of characteristics can be potentially infinite and enormously detailed. It can be subdivided in six macro groups as: **User kind**, i.e. citizen, tourist, commuter, operator, student; **User info**, i.e. gender, age, nationality, spoken languages, special needs, velocity, mean of traveling, acceleration; **Place/location**, i.e. Global Positioning System (GPS) coordinates, municipality, region, nation, terrain topology, area description, population of the area; **Date/Time**, i.e. day, date time, day of the week, holiday day, season, month, year, temporal window (from-to date); **Environment/sensor**, i.e. weather condition, level of water in a river, pollution, temperature; **City context**, as *closest to* (i.e. a Point Of Interest 'POI' with its description, street, cycling path, railway, highway) or *inclusion into* (i.e. parking, fuel station, green area).

The values assumed by a $C_i(u, t)$ can be represented as a number, free text, date/time/duration, enumerates, as well as complex structured data values. The sequence of values $C^{ts,te}(u)=\{C(u, ts), C(u, ts+1), … C(u, te)\}$ describes the evolution of each User Profile $C(u)$ in an interval of time [ts,te] and can be visualized as in **Figure 2.**
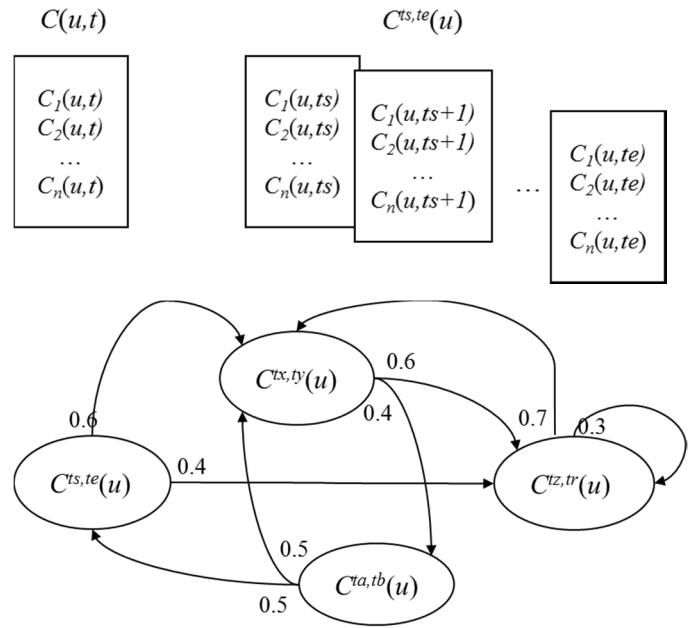

Figure 2 - User Profile Evolution

The set of User Profile Evolution include also similar descriptors for collective profiles, and for the typical User Profile Evolution: the typical User Profile Evolution. Thus, a Markov Chain of probable status with related transition is constructed among possible, $C^{ts,te}(u)$, representing data from the past, and possible predicted status for the future and transitions [18], [19].

B. User Behavior Data Anaytics

Many of the above mentioned characteristics $C^{ts,te}(u)$can be directly obtained from the mobile phone data. Others, , can be computed from the contextual data by the User Behavior Data Analytics. For example, the GPS coordinate may provide information about the region and nation in where the user is located in, the temperature for that location may be known, as well as the weather status. In general, a set of processes/formulas $F(C^{ts,te}(u))$ may be set up and computed by using:

- semantic computing: SPARQL Protocol and RDF Query Language (SPARQL) queries exploiting inference on the Km4City knowledge base by exploiting spatial and temporal reasoning. For example, to know the closest POI, the inclusion in an area, the weather, the next busses arriving at the bus stop, the entertainment events in the city in a given time slot, etc. [6].

- Data analytics: machine learning, clustering and/or statistical approaches based on collected characteristics in the time interval as well as in the past. For example, by creating algorithm capable of computing/predicting

- the means of traveling (car, bike, bus, etc.) can be derived from the velocity, acceleration, proximity to bus line path, etc.

- the density of people, thus minimizing/maximizing the probability of meeting other people [7],

- the traffic flow and/or parking in other areas may be known, while may be predicted for the future, etc.
- the next possible target location for each given user according to his/her personal story.

The evaluation of $F(C_{ir}(x)(u))$ may be time consuming. For most of the feature can be computer offline for many different user at the same time, and storing the results in common storage and knowledge base, or in the User Behavior Evolution if the value is specifically associated with that user. In Sii-Mobility, most of the computational activities assessing the conditions are scheduled as periodic background processes and managed by the Distributed Smart City Engine Scheduler (DISCES).

C. Action and Eng. Msg. Computation

An action A is defined as a box container for an Engagement Message (EM), ruled and enriched by a list of characteristics $A_1, A_2, ... A_m$, representing the modality of delivery of *EM* for a given user *u*:

$$A(u) = \{A_1, A_2, ... A_m, EM(u)\}$$

Example of action characteristics can be the sending rate of a particular type of engagement or the maximum number of engagements to be delivered simultaneously, or provided in a time frame, according to strategies.
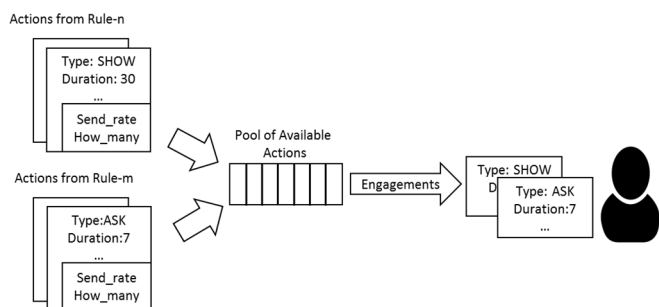

Figure 3 - Action pooling to engagement delivery

The UEEngine takes care of the management of the pool of Actions prepared to be sent (fired rules upon condition's validation) and to deliver the choose ones according the specified modality (i.e., to avoid user flooding and to fine tuning, the orchestrator can decide that some type of engagement should be delivered just every hour). The scenario is sketched in the Figure 3.

The computation of an Engagement Msg, *EM(u)*, is performed based on the Strategy defined by the Operator by putting together several different elements of information to be delivered to the Mobile Application, depending on the Engagement kind. An engagement *EM* is defined as a list of elements $EM_1, EM_2, ... EM_n$, delivered to the user's mobile:

$$EM(u) = \{EM_1, EM_2, ... EM_n\}$$

The elements of an Engagement Message can include:

- type:

- SHOW: to inform by showing – e.g., alert of civil protection, weather forecast, if you parked here please remind to take the ticket;
- ASK: to request an action – e.g., take a photo, provide a comment;
- SWITCH: to ask confirming an analysis – e.g., 10 minutes ago you moved via: car, bus, bike?

- target destination of the engagement: it may be user via Mobile Application interface, or could be sensor/display in the user's proximity;

- deadline: time duration of the engagement into the Personal Assistant page and notification in the Mobile App. It is usually expressed as a time out over that the message is removed;

- additional data: details for the action, i.e., a string text message in case of an engagement of type SHOW.

D. Feedback Rules definition

The set of Feedback Rules are defined to provide to the user a flow experience (feedback loops) in term of rewarding schemas (i.e., providing discount, offer bonus, manage fidelity card) and gaming experience (i.e., compile a leader board, promote challenges).

The Feedback Rules involve the definition of strategies that strongly depend on the target scenario of the integration. The engine, offers a pluggable runtime system for defining monitoring tools (one for any type of engagement), that analyses the user behavior and upon the observation that a user has been actually engaged in some kind of activities (after an engagement has been received and properly consumed) thrown a signal that can be catch by a listener that implement a feedback rule. The listener can specify a timeout, as the maximum amount of time to wait until the engagement is feedback.

A set of basic feedback rules are automatically loaded by the platform. They take care to update explicitly the user's context with the observed fact and to enable analysis and discovery of user's behavior pattern similarities.

IV. IMPLEMENTATION NOTES

The UEEngine has been realized in the context of Sii-Mobility project and thus integrated in the general framework which includes the Km4City knowledge base and tools. In the context of that project UEEngine has been integrated according to Figure 1, and providing specific Smart City API towards the mobile applications. Thus a couple of mobile applications have been implemented and are available on Google Play Store, Apple Store, Windows Market, and Windows 10, named as "Firenze, where, what,..", "Toscana, where, what,..", and supporting 5 different languages (ITA, ENG, ES, DE, FR).
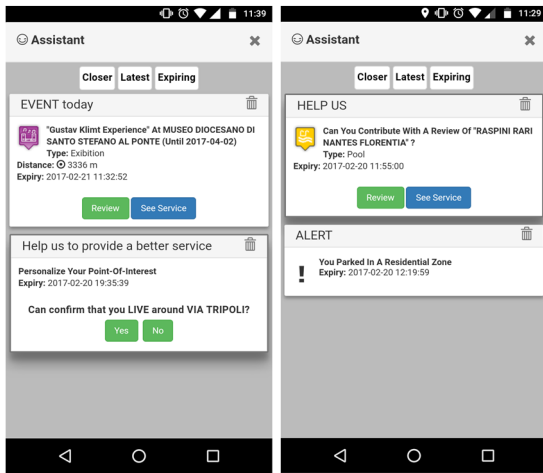
Figure 4 - Engagement on Mobile Application

The UEEngine has been realized by using JAVA by using MVFLEX Expression Language (MVEL) and/or the Friendly Enough Expression Language (FEEL). The MVEL is a hybrid dynamic/statically typed, embeddable Expression Language and runtime for the Java Platform. It's currently supported in several platforms and stable enough to have some graphical presentation (DROOLS workbench) [20]. The Droolsbox tool has been used to execute the Engagement Rules and their triggering.

V. EXPERIMENTAL RESULTS AND VALIDATION

The UEEngine has been used to stimulate the city users to have a more sustainable mobility, to contribute at the information collected on the knowledge base, to improve the mobile application and to have more attendance on events.

Specific User Behavior Data Analytic processes have been implemented to assess:

- closeness of a point, over bus line, train line, highway, cycle paths, and inside areas, as parking, parks, …;

- user mobility mode: analyzing the user's device context characteristics (speed, acceleration, GPS, …) and the transport public network, determinate if the user is: still, walking, running, driving, biking, using the motorbike/tram/bus/train;

- change of mobility mode: deriving the dynamic evolution of the user mobility mode and the transport public network determinate if the user is: parking the car, taking the car, taking the bus, …;

- identify the Personal POI (PPOI): analyzing the user mobility mode in specific day-slots, determinate the GPS coordinates of the user's home, the user's place of work or other PPOIs [21];

- identify the habits for a number of users in moving among their PPOI: work-work, home-work, work-PPOI, etc [22].

A number of engagements to push to the city users have been formalized: ask the user to fill a survey (one-shot question, detailed survey); ask the user to comment a POI (take a photo,

comment, rate); notify the user a message (an alert, a description message); notify the user an event (its coordinates, its duration).

```
rule "parking_en"
  dialect "mvel"
  when
    ppoi:PPOI(name == "HOME")
    USER(languages contains "en", switchMobilityMode ==
      "PARKING", ppois contains ppoi)
    LOCATION(cpz != ppoi.location.cpz)
  then
    ACTION fact0 = new ACTION();
    fact0.setType("SHOW");
    fact0.setTitle("ALERT");
    fact0.setMsg("you parked in a residential zone");
    fact0.setTime_elapse(30);
    fact0.setAction_sendrate(0);
    insert(fact0);
end
```
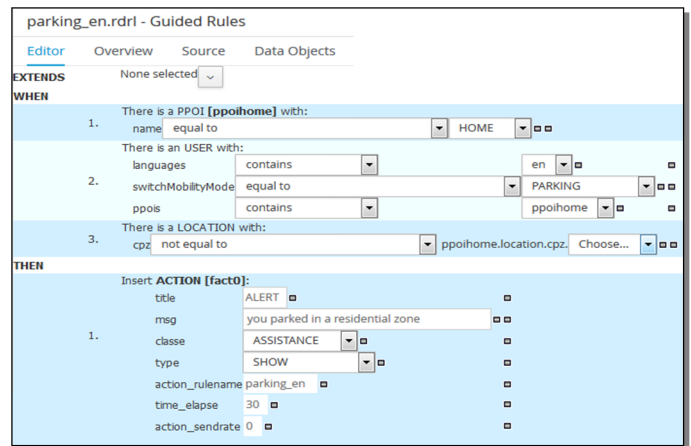

Figure 5 - Example of Engagement Rule

With these lists of conditions and actions a total of 45 rules have been put in place. For example, in **Figure 5**, a rule to SHOW an ALERT in English on the device when the engine detects that the city user is parking in a controlling parking zone different from that or his residence is presented, by using MVEL language and on Drool workbench tool.

The UEEngine, with the above-mentioned rules is up and running for validation since the 1st of September 2016. It has been able to produce 52.708 engagements on a total number of 2363 users (belonging to different categories, citizens, tourists, etc.). About 70.000 daily users' profiles have been analyzed.

The execution of the User Behavior Data Analytics on the citizens detected the HOME PPOI locations (in which the city user pass at night several hours) for the 89.43% of users, the WORK PPOI location for the 65.21% of users. The tracked temporal window was set at 8 weeks and thus, users with less than 20 recurrent positions have been discarded as well as locations with low GPS accuracy less than 100m. The execution time for the whole analytics was put in execution every 6 hours to keep update the lists of PPOIs. During the

validation test with a restricted number of 15 users from the lab, the percentage of precisely classified PPOI as HOME or WORK has been of the 93.4%.
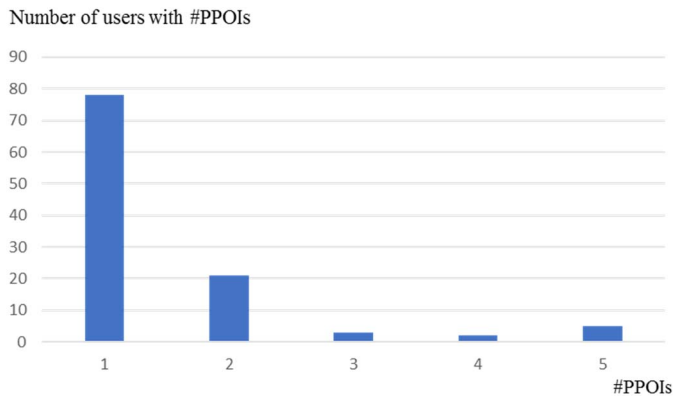

Number of users with #PPOIs

Figure 6 - Distribution of number of city users with additional PPOIs with respect to those classified as HOME or WORK

Moreover, in the temporal windows from 2016-12-12 to 2017-2-6 (8 weeks) the mobile Apps have been put in the hands of the final users detecting a total of 344 Personal POIs on 200 users. Among them, 182 PPOIs have been classified as HOME or WORK. Plus, 162 PPOIs on 109 city users have been identified having different classification with respect to HOME or WORK. The **Figure 6** presents the distribution of number of city users with additional POIs with respect to those classified as HOME or WORK, identified in the same timeslot.

In the same 8 weeks, it was possible to compute the most frequent trajectories of the users between their PPOIs has been analyzed. A total of 3588 trajectories has been identified for 84 different users. A clustering based on the day of the week and the slot of the day (night, morning, launch, dinner, evening) has been applied. Among the above mentioned trajectories 1799 presents a number of repetitions as reported in the **Figure 7**. These trajectories can be classified according to their relationship with HOME, WORK or X (other PPOIs) as reported in **Figure 8**.
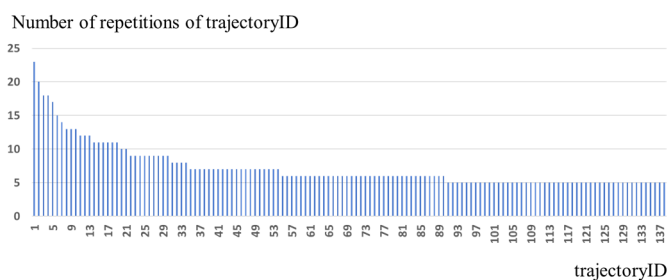

Number of repetitions of trajectoryID

Figure 7 - Distribution of repetition for trajectories IDs

All these 3588 trajectories have been used to construct a Hidden Markov chain for each user: the probability of a user move from a PPOIs to the next, in the weekday and time slot

the observation. The prediction model was tested in the timeslot [7-2-2017, 14-2-2017]. During the validation predictions have been delivered only when the Markov probability was above the 65% (same days, day-slot, starting PPOI, etc.), and the prediction was delivered within 3 minutes from the start. Thus the 76% of predictions have been proved to be correct, observing the arrival at destination predicted PPOI.
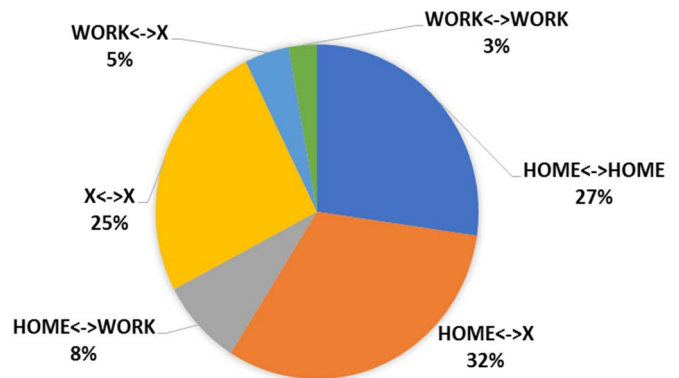

Figure 8 - Distribution of different trajectories

As a final consideration, the UEEngine also delivered a number of simple engagements such as those to simply inform and stimulate the city users against facts. In this context, we observed that the rate of engagements viewed against the engagement sent strongly depend on the time validation for the engagement (the permanence duration of the engagement into the mobile phone of the city user). Even some the same kind of engagements, those remaining available for 30 minutes have the 1.9% of views, while those remained active for 12 hours have been viewed 30.15%. This measure has been possible instrumenting the mobile app.

VI. CONCLUSION

The new challenges in the smart city context are mainly related to the stimulation of the city users towards sustainable behaviors. For example, in mobility, exploiting energy, using city services. On this context, many ad-hoc solutions have been proposed for informing the city users and/or for engaging them with specific wired rules toward virtuous models. In this paper we proposed a flexible language, exploiting analytics and predictive models, which can be used to identify users that need to be pushed towards a range of virtuous habits. On this regards, the main problems are (i) the computation of user behavior via data analytic (semantic computing, machine learning), as well as (ii) the formalization of strategies via simple and well defined language and tools for producing engagements to the city users. The language has to usable by city operators which are those that put in place the strategies on the input of political parties. In this paper, the solution for city users engagement studied and implemented for Sii-Mobility Smart city national project in Italy has been presented. The solution has been implemented thanks to the exploitation of Km4City model and semantic computing, and a number of data analytics algorithms. The paper also reported the first results in

assessing user behavior and in using the engagement rules, putting in evidence some aspects, which can be exploited for setting up more complex rules in the future.

REFERENCES

[1] CKAN, the open source data portal software. [Online]. Available: http://ckan.org. [Accessed: April 7, 2017].

[2] CitySDK, City Service Development Kit. [Online]. Available: http://www.citysdk.eu. [Accessed: April 7, 2017].

[3] Transport.api, the digital platform for transport. [Online]. Available: http://www.transportapi.com. [Accessed: April 7, 2017].

[4] OpenDataSoft. [Online]. Available: http://www.opendatasoft.com. [Accessed: April 7, 2017].

[5] P. Bellini, D. Cenni and P. Nesi, "Optimization of Information Retrieval for Cross Media contents in a Best Practice Network", International Journal Multimedia Information Retrieval, 2014, pp. 147-159.

[6] C. Badii, P. Bellini, D. Cenni, G. Martelli, P. Nesi and M. Paolucci, "Km4City Smart City API: an integrated support for mobility services", IEEE International Conference on Smart Computing (SMARTCOMP), 2016, pp. 1-8.

[7] P. Bellini, D. Cenni and P. Nesi, "AP Positioning for Estimating People Flow as Origin Destination Matrix for Smart Cities", International Conference on Distributed Multimedia Systems, DMS, Italy, 2016.

[8] F. Giannotti and D. Pedreschi, eds. "Mobility, data mining and privacy: Geographic knowledge discovery", Springer Science & Business Media, 2008.

[9] D. O'Sullivan, B. Smyth, and D. Wilson, "Understanding case based recommendation: A similarity knowledge perspective", International Journal on Artificial Intelligence Tools 14.01n02, 2005, pp. 215-232.

[10] G. Cardone, et al. "Fostering participaction in smart cities: a geo-social crowdsensing platform", IEEE Communications Magazine 51.6, 2013, pp. 112-119.

[11] H. L. O'Brien and E. G. Toms, "What is User Engagement? A conceptual framework for defining user engagement with technology", Journal of the Association for Information Science and Technology, 59(6), 2008, pp. 938–955.

[12] S. Attfield, G. Kazai, P. Benjamin and M. Lalmas, "Towards a science of user engagement", WSDM Workshop on User Modelling for Web Applications, Hong Kong, China, 2011.

[13] J. Lehmann, M. Lalmas, E. Yom-Tov and G. Dupret, "Models of User Engagement", User Modeling, Adaptation, and Personalization, Vol. Lecture Notes in Computer Science, 2012, pp. 164-175.

[14] I. G., Mart, et al. "Mobile Application for Noise Pollution Monitoring through Gamification Techniques", M. R. Herrlich M., Entertainment Computing, ICEC 2012, Vol. Lecture Notes in Computer Science (LNCS), 2012, pp. 562-571.

[15] D. Banister, "The sustainable mobility paradigm", Transport policy 15.2, 2008, pp: 73-80.

[16] M. Csikszentmihalyi, "Flow: The Psychology of Optimal Experience", 1990.

[17] B. Hu, T. Paktos, A. Chibani, and Y. Amirat, "Rule-based context assessment in smart cities". 6th international conference on Web Reasoning and Rule Systems, 2012, pp. 221-224.

[18] Z. Ghahramani, "An introduction to hidden Markov models and Bayesian networks", International Journal of Pattern Recognition and Artificial Intelligence, 15(1), 2001, pp. 9-42.

[19] N. Novielli, "HMM modeling of user engagement in advice-giving dialogues", Journal on Multimodal User Interfaces, 3, 2010, pp. 131-140.

[20] DROOLS Business Rules Management System. [Online]. Avaialable: https://www.drools.org. [Accessed: April 7, 2017].

[21] D. Ashbrook, T. Starner, "Using GPS to learn significant locations and predict movement across multiple users". Personal and Ubiquitous Computing, 7(5), pp. 275-286.

[22] Y. Zheng, L. Zhang, Z. Ma, X. Xie, and W.Y. Ma, "Recommending friends and locations based on individual location history", ACM Transactions on the Web, 5, 2011 1-5:44.