# Delivering IoT Smart Services through Collective Awareness, Mobile Crowdsensing and Open Data

Federico Montori*, Luca Bedogni†, Gianluca Iselli‡, Luciano Bononi*
*‡Department of Computer Science and Engineering, University of Bologna, Bologna, Italy
†Department of Physics, Informatics and Mathematics, University of Modena and Reggio Emilia, Italy
*{federico.montori2,luciano.bononi}@unibo.it, †luca.bedogni@unimore.it, ‡gianluca.iselli@studio.unibo.it

***Abstract*—IoT is spreading heavily in many use cases that surround our everyday life; however, existing IoT ecosystems are still behaving as close islands with little interoperability with each other. Recent research efforts tend to propose new architectures and standards to which private customers and companies producing data are supposed to adhere in order to make them consistent. However, such entities have their own vested interests that hinder data integration. We instead leverage information acquisition through Collective Awareness Paradigms (CAPs) such as Open Data and Mobile Crowdsensing (MCS) in order to use what is already available. As a proof of concept, we developed SenSquare, a prototype IoT architecture and platform for Smart Cities and environmental monitoring that gathers raw data through CAPs, adapts it to a common semantic and composes customizable flexible services. Inexperienced users can generate their own services using an easy visual programming plugin designed around a customized language. Furthermore, we test the platform on a real world use case.**

***Index Terms*—IoT, Open Data, Mobile Crowdsensing, SOA.**

## I. Introduction

The spread of the Internet of Things (IoT) in its multiple forms has been impressive over the last 15 years. The research in this direction has been largely fostered, also because the areas of research are by now numerous; it is indeed impossible to give a unique definition of how an IoT ecosystem is ought to be structured. In fact, many of the use cases have orthogonal features and the areas of research pertaining the IoT are getting farther and farther from each other. This is evident, since nowadays many of the IoT-related works in literature highlight the growing number of connected devices and the amount of billions of dollars invested in the IoT market [1], together with the plethora of technologies, standards and protocols that have been proposed and distributed in the market. The outcome of this consideration is that this universe is currently moving at a pace that sometimes research and standardization efforts cannot keep up; as new needs come in, industrial ad-hoc efforts tend to come first. In fact, we are currently witnessing a set of "Intranets of Things" rather than a true Internet of Things [2], this is because current ecosystems tend to behave as closed islands with little or no interoperability between each other. This happens either because solutions need to be deployed "here and now" – so there is more need for the least solution that solves the problem rather than something that can someday be useful to others – or else because widespread and different industrial solutions force the customer to stick to what has been envisioned by the manufacturer. Of course, architectural and semantic standardization efforts have been envisioned. As a matter of fact, the literature is literally covered in proposals for new IoT frameworks and architectures expected to cover a plethora of use cases; however, many of them end up being yet another standard that a small amount of solutions adopt. The prosperity of a standard has to be examined from the viewpoint of whether the participants achieved their goals from their participation in the standardization process [3]: in our case, the various parties have much at stake in their own interests, therefore standardization efforts in some IoT fields find a lot of obstacles.

As opposed to the situation described above, we leverage the concept of Collective Awareness paradigms (CAP) [4], a revolutionary way to think about IoT: instead of proposing a new standard, the priority is to adapt the architecture to what is already in place through a wise use of a collective effort. CAPs rely on the cooperation of participant users that willingly share their data through public repositories to be used and aggregated by third parties in order to construct a common model. For a number of use cases based on common resources (i.e. environmental monitoring and Smart Cities) such paradigms have demonstrated to be a key enabler to a whole new level of pervasiveness for any application as well as a significant reduction in the costs. In fact, what was formerly required – i.e. ad-hoc deployments of sensor networks – is, for many applications, no longer necessary. Clearly, this paradigm is characterized by different issues that formerly were merely technological and now shifted to social: people need to be instructed, encouraged to participate, incentivized and satisfied of the results. The greatest outcome that CAPs bring to the current trends is given by three major concepts:

1) Data about phenomena of common interest is probably already in place. The more the interest, the more likely is to find Open Data that can describe it, being it institutional (i.e. gathered through reliable appliances) or crowdsourced.
2) If data is not available, devices capable of reporting observations about such phenomena are probably already in place. Even though Mobile Crowdsensing (MCS) is the biggest trend that exploits this concept, any collaborative

solution could make the difference in such sense.
3) The need for IoT services does not come only from companies, but also from citizens. They need a simplified and accessible way to design their own customized services.

With the final goal of bringing interoperability to IoT ecosystems and bridging the gaps of data and device redundancy, we illustrate our vision though an implemented prototype architecture, called SenSquare, which leverages the usage of CAPs (crowdsourcing and crowdsensing), aggregates data from heterogeneous sources and delivers to the final user a customized and straightforward way to monitor and react to phenomena in a service oriented fashion. That being said, we do not aim to propose yet another standard, but rather we aim to make the most out of what is already in place and present a solution that outlines our vision on the Collaborative IoT. In short, the paper brings the following contributions, also with respect to previous related publications [5]:

1) Different CAPs (MCS and Open Data) sources are analyzed and integrated together in a common paradigm and data is processed and classified whether necessary.
2) A more expressive dedicated language for service composition is proposed and defined.
3) A novel visual programming way to define customized services for end users is proposed and implemented.
4) A translator for service templates into Python scripts, in order to execute them as blackboxes, has been proposed and implemented.

The paper is structured as follows: Section II introduces the different facets of CAP and our motivation for focusing on them, Section III introduces SenSquare and discusses the architectural details and its elements, Section IV presents the user experience of our platform, finally, Section V concludes the paper.

## II. Collective Awareness: the Collaborative IoT

A fundamental building block of the present work is given by the CAPs, a set of methodologies and systems that leverage the collaboration in data collection and other fields in which a complex and resource-consumptive task is offloaded to a crowd of participants. This results in a minimal effort for the individual, a benefit for both the executors and the issuers and a massive economic saving. Our vision supports a paradigm in which services are delivered through what is already in place, leveraging cooperation between final users and stakeholders. Specifically, we make use of two major paradigms: Open Data and Mobile Crowdsensing (MCS).

### A. Open Data

Open Data is, as the name suggests, data that is freely accessible in machine-readable format from public repositories. It may be either contributed by users or gathered in an open access form through an initiative. In fact, we group Open Data repositories as "reliable", that is, repositories maintained by organizations or governments, and "unreliable", that is, repositories created through crowdsourcing: users freely contributing in uploading IoT data through their personal devices [5]. Reliable Open Data repositories are preferred, since the data they provide follows some sort of annotation policy (i.e. we know exactly what it is), is updated regularly and its quality is guaranteed by the use of professional appliances. Examples of reliable Open Data repositories are the Environmental Protection Agency (EPA)[1], providing environmental monitoring data in the United States, and various weather and forecast services such as DarkSky[2]. Unreliable Open Data repositories, on the other hand, provide IoT data without any warranty about its veracity, neither about what it actually measures. Data is typically unlabeled, poorly annotated and incomplete and needs a processing step to classify which data feeds are valuable and what they actually measure. An example of crowdsourced unreliable Open Data repository is ThingSpeak[3], a platform where users can upload data generated by their personal devices (mostly environmental) in "data channels" and make them public.

Why then unreliable Open Data repositories should be of interest? First of all, it is worth noting how data coming from reliable sources is provided at a wide area granularity (often per-city), which may be inaccurate for some applications. An example is the acoustic noise, which varies dramatically in small distances in time and space. Another reason lies upon the general trend in the usage of these platforms throughout a time window of few years. Let us consider the example of ThingSpeak, for which we analyzed all the public data channels (around 160.000 out of a total of 600.000 are public at the time of writing), all of them coming with a creation date and the time of the last update. We report the channels in the diagram in Figure 1. For each month in the diagram, the horizontal line inside the boxplot represents the number of active channels, the upper box is the number of newly created channels, and the lower box is the number of channels that have been updated for the last time on such month (we assume them to be no longer active since then). Green boxes are those for which the number of created channels is higher than the number of channels that ceased their updates, red boxes are the opposite.

We can observe a global increase in the number of active data streams since 2011, as almost every month shows an increase in the number of active channels. Motivations behind this phenomenon are the reduction of the cost for components such as Arduino, ESP8266 and alike. Moreover, related tools are now much easier to use, reducing the digital gap and flattening the learning curve to develop embedded software. Furthermore, it is interesting to observe how the integration of heterogeneous sources would improve the knowledge base, due to their specialization in different observation fields as well as their different geographical concentrations. Such analyses shed some light on how rapidly the world of Open Data is growing and people are gaining interest in using a platform

[1] https://www3.epa.gov/
[2] https://www.darksky.net/
[3] https://thingspeak.com/

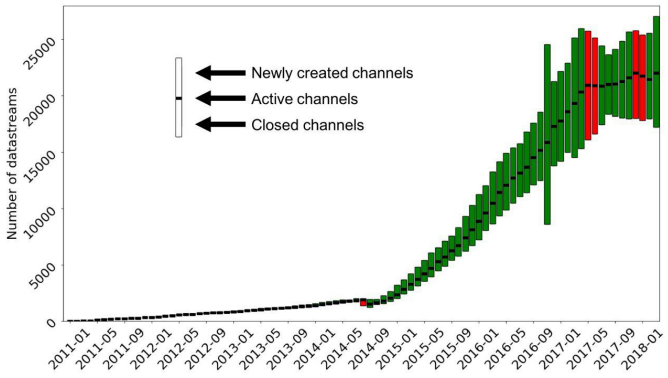



Fig. 1. Trend in creation and update of ThingSpeak channels.

that takes away the burden of creating a local ecosystem.

### B. Mobile Crowdsensing (MCS)

Mobile Crowdsensing (MCS) is a CAP coined in [6] and defined as *"a paradigm through which a number of individuals, called participants, are committed to perform tasks – as part of a campaign – involving sampling of real world phenomena of common interest through the use of portable, connected and Spatio-Temporal Aware mobile devices in order to enable its mapping through information aggregation"* [4]. The definition includes any connected mobile device capable of observing phenomena and performing computation, however, MCS refers predominantly to Smartphones. In [6] MCS was first classified into participatory and opportunistic, a separation that has been pointed out in other subsequent works. In particular, Participatory Crowdsensing is a paradigm in which the user is actively involved, often through the use of a front-end application, and intentionally reports observations through a specific action, whereas in Opportunistic Crowdsensing the user involvement is minimized and often an application is running in background performing sensing, monitoring tasks and performing decisions on where and when to sense and send on behalf of the user. MCS yields a massive data gathering at a significantly low price, although it presents a multitude of challenges that are addressed separately in literature [7], such as data quality [8], incentivization techniques [9] and recruitment algorithms to deal with scenarios in which data can be too sparse or too dense for the purpose of the application [4]. We consider MCS mainly for the purpose of Smart Cities and environmental monitoring, areas in which it has been greatly leveraged.

## III. SenSquare: Architecture and Modules

The effectiveness of our paradigm is demonstrated through the real implementation of our prototype platform SenSquare. It has first been introduced in [5] and, since then, we have explored all its components. The overall architecture is presented in Figure 2.

In the figure, we can see clearly three different sections:

1) The data gathering part, devoted to collect data using CAPs.
2) The data aggregation part, committed to unify data coming from different sources and forming a layer of abstraction that transforms raw data in complex services.
3) The service domain, in which users create, share and make use of customized and dedicated services.

### A. Data Gathering

The lowest layer of the architecture of SenSquare is in charge of retrieving useful IoT data from publicly available resources. This task is clearly non-trivial due to the heterogeneity of the data sources as well as their potentially variable data quality. Within the scope of public Open Data, the only way to retrieve all the possible data feeds is to construct a dedicated data scraper that periodically performs HTTP requests in order to extract the updated data points from the web. Both reliable and unreliable resources have been tested with success, in particular, we extracted air quality data from the Regional Agency for the Protection of the Environment in the Italian region Emilia-Romagna (ARPAE)[4]. Such data is annotated and its precision is ensured by the quality of the appliances. For the unreliable data sources, we extracted the whole knowledge base of the Open Data clouds of ThingSpeak and SparkFun[5] (although its public cloud is not available anymore since 2017), since they present similar issues in terms of annotation. In fact, the crowdsourced data feeds in ThingSpeak are not properly annotated (i.e. the measurement class is not explicitly stated), thus, we potentially do not know what is being measured. For such reason, in order to automatically classify each feed (defined as "datastream") we proposed a sequential ensemble algorithm that combines classifiers for both numerical and natural language data [10]. The algorithm has been tested on a number of datasets, among which the ThingSpeak dataset that we produced and made openly accessible[6], and it is shown to outperform canonical approaches in literature such as [11].

We also leveraged MCS as a powerful source of information for our framework. In particular, we focused on the opportunistic collection of data, i.e. we built a mobile Android application that periodically uploads data collected by the sensors in the device. Each type of data has its own spatial and temporal ideal granularity at which it has to be sampled. This has been implemented through a set of rules that are fine tuned through a self-adaptable distributed probabilistic algorithm. The goal of the algorithm is to regulate the amount of collected data to an established optimum and to preserve the privacy of the users. In particular, it pushes the participant devices in contributing more when data is too sparse in the area and it limits data uploads when data is too dense [12].

### B. Data Annotation

In order to step into the stage of service composition, semantic annotation is essential to uniformly access heterogeneous IoT data. In order to fill such gap we envision a tool that

[4]https://www.arpae.it/
[5]https://www.sparkfun.com/
[6]https://github.com/stradivarius/TSopendatastreams

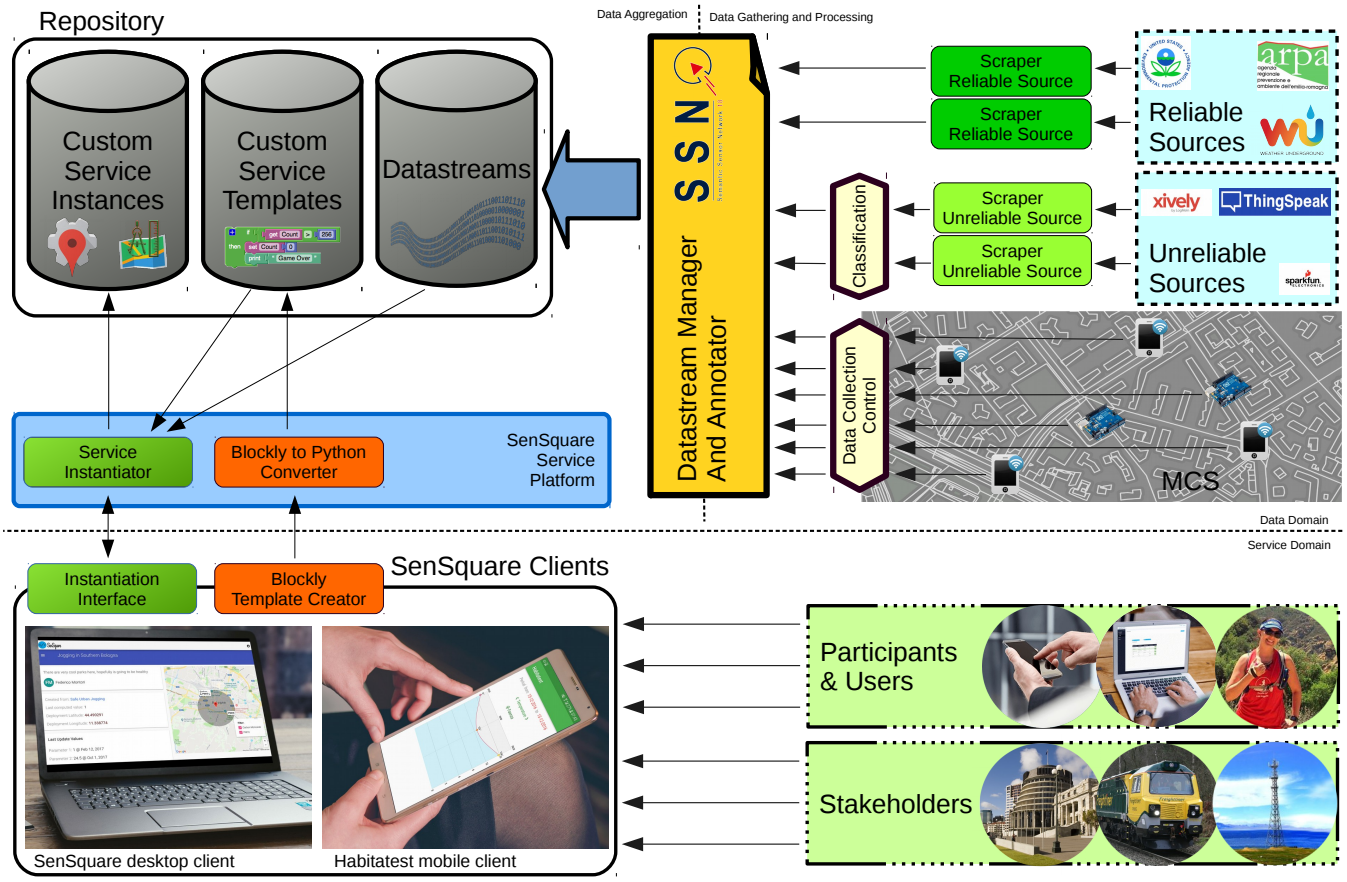



Fig. 2. Architecture of the whole ecosystem together with all of its external and internal components and actors.

is able to semantically enrich IoT data with metadata extracted from semantic sensor ontologies. In particular, such platform, called "Datastream Manager and Annotator (DMA)", is included as part of the architecture in Figure 2 between the data gathering environment and the service part. Such tool gathers data from all the data gathering and processing components and saves it into the repositories using a unique format. It could be easily extended to an automatic datastream annotator using metadata extracted from domain-specific ontologies – i.e. ontologies that are specific to the data type, for instance the air quality – and semantic sensor ontologies such as SSN [13] or SOSA [14], using as input the data class received from the gatherers or, when missing, the output of the classification algorithm. In such a way the DMA would be able to associate the SSN or SOSA class with the output class from the classification module, then we expect it to semantically enrich the annotated IoT datastreams with domain specific concepts provided by the IoT application. An accurate implementation in such direction is currently under study.

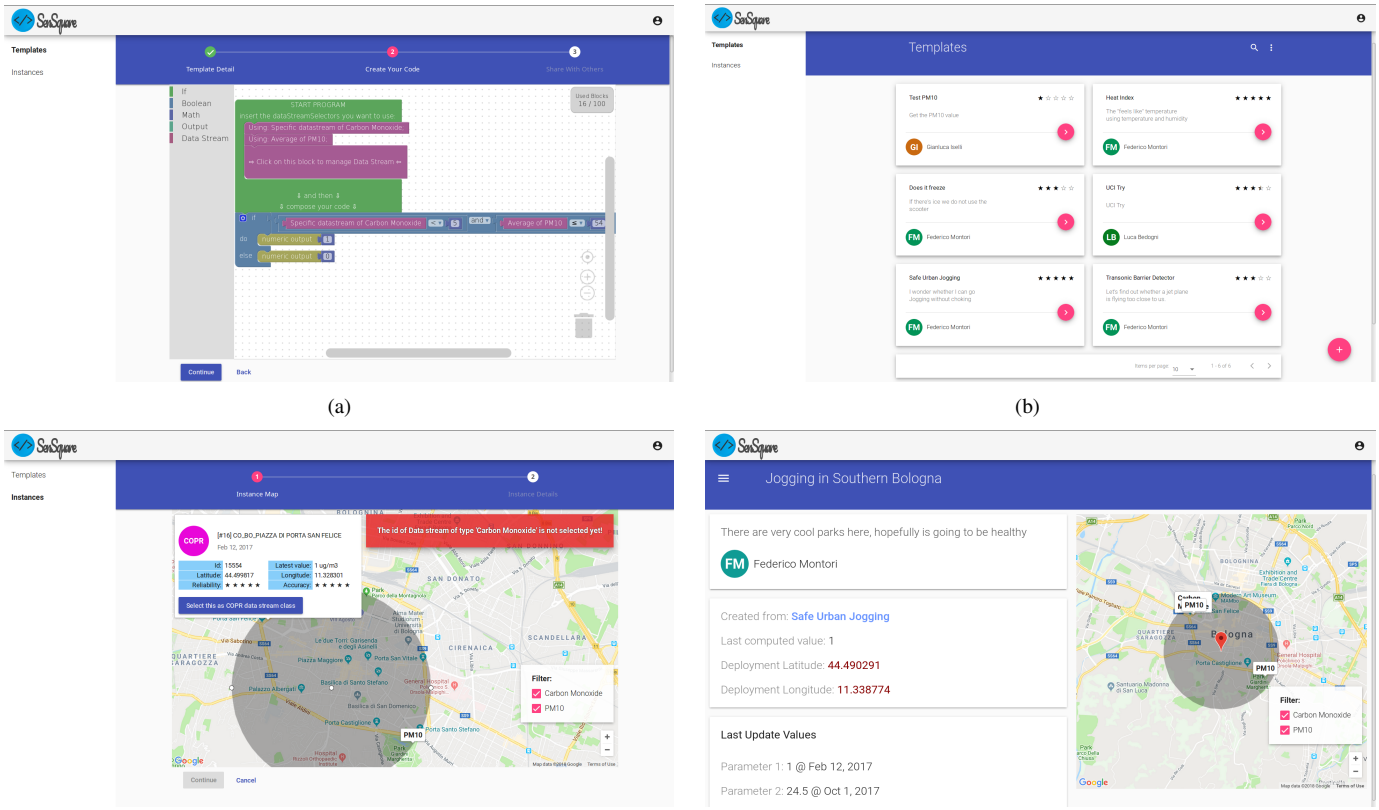
### C. Service Delivery

Service Oriented Architectures (SOA) are the added value to pure IoT applications, since they leverage the service composition of raw data and add reasoning capabilities, making observations much more meaningful to humans. In our case, services are composed through two main primitive entities: the datastreams and the Custom Service Templates (CST). CSTs are designed by the users and shared in a common repository to encourage reuse. They are abstract composition of primary types of measurements and users design them in the same way a programmer writes a function. CSTs are designed using a dedicated language, defined here in Backus-Naur Form (BNF), that includes basic arithmetic and relational operations between datastreams, the `if-then-else` clause and logical connectives. Formally, a CST is defined by a mathematical expression $E$ as follows:

$$E := c \mid DC \mid (E + E) \mid (E - E) \mid (E * E) \mid (E/E) \mid IFTE(C, E, E)$$

$$C := b \mid C \wedge C \mid C \vee C \mid \neg C \mid E > E \mid E \geq E \mid E < E \mid E \leq E \mid E = E \mid E \neq E$$

where $c$ is a constant floating point value, $b$ is a boolean value and $DC$ is a datastream class. $IFTE(C, E_1, E_2)$ is the `if-then-else` clause, which executes $E_1$ if $C$ is true, $E_2$ otherwise. When defining each $DC$, the CST specifies whether it should correspond to a single datastream; in alternative, aggregated measures for all the datastreams of the same type can be used (i.e. the maximum, the minimum and the average).




(a) (b) (c) (d)

Fig. 3. Screenshots of the main functionalities of the SenSquare Web application.

A CST is then stored in the database as a Python script with the used datastream classes as parameters and it is executed as a blackbox. Given such primitives, the actual services are defined as Custom Service Instances (CSI), again generated by the users through the instantiation of a CST in a specific geographical area. When a CSI is instantiated, the user can choose the specific datastream of each type located in the area of interest and required by the respective CST to be used for the calculation. In alternative, aggregated measures for all the datastreams of the same type can be used (as specified by the CST definition). After this stage, the CSI behaves as a new datastream, which takes in input raw measurements and returns periodically a numeric output, using the expression contained in its CST as a calculation function. CST and CSI repositories are public, thus, once they are created, they are accessible to all the users of the platform. For the sake of clarity, in the next section we give an example of usage for CSTs and CSIs.

## IV. SenSquare: the User Interface

In this section we present the current SenSquare desktop client application[7], where users make use of the datastreams gathered through collective awareness by creating CSTs and instantiating them into CSI. In order to better explain the usage of the platform we will walk the interested reader through an example that better clarifies each step. Let us say that a user is particularly sensitive to urban pollution, however, he or she is also interested in jogging in zones included in the urban area. In such case, the user would start with the creation of a CST that informs whether the outdoor air quality is good enough to preserve her health. Looking at the EPA air quality indices (AQI), we can establish the maximum level of PM10 (suspended particulate matter below $10\ \mu\text{m}$) bearable for a good AQI is $54\ \mu\text{g}/\text{m}^3$, whereas the maximum level of CO (Carbon Monoxide) is $5$ ppm. Hence, we would write a CST that, if both the levels are below the respective thresholds, would output a positive value, negative otherwise. We do not assume that inexperienced users have programming capabilities, therefore, we leveraged the paradigm of visual programming, widespread in the field of education, for the composition of a new CST. In particular, we used the well-known plugin Blockly by Google[8] with customized functionalities in order to cover the only the cases outlined in Section III-C (i.e. avoiding loops) and provide as variables only data classes. Whenever selecting a possible data class that can be part of the CST, we ask to the user whether it has to be a specific value or an aggregate during instantiation. In our example, the composition of the CST through Blockly is depicted in Figure 3(a), in which new blocks can be dragged and dropped from the left end side into the main dashboard.

[7]http://sensquare.disi.unibo.it/

[8]https://developers.google.com/blockly/

We set the value of PM10 to be an aggregate (the average value), whereas the value of CO has to be selected from a specific datastream at the time of instantiation. Once the CST is generated, it is stored together with all the other CSTs created by other users. The list of CSTs is shown in the screen in Figure 3(b), where users can explore all the created CSTs and select to instantiate one of them. Given that CSTs are created through crowdsourcing, we introduced a rating mechanism in order to quantify the trustworthiness. Once the user selects one of the CSTs from the list, she will be displayed with the instantiation wizard screen, which consists in a map where the user should indicate a circular zone of interest (both the center and the radius are customizable). As the user moves and edits the circle, all the static and active datastreams within such area are displayed with a marker on the map (the datastreams coming from MCS sources are not displayed, since they are moving constantly and, therefore, will only take part in the aggregates). Only the datastreams of the same classes as the ones required by the respective CST are shown, in our example only datastreams measuring PM10 and CO. In Figure 3(c), following our example, we need to select a single CO datastream, whereas for PM10 it is not necessary, as we are using the average over all the PM10 datastreams in the area. Once the CSI is created it will be available to the whole community to be visualized. In Figure 3(d) we show the CSI visualization screen for our instantiated example. On the right side, the map with the circular area highlighted in dark grey is displayed, together with the markers representing all the static sources taken into account. It is also possible to filter them by type. The part on the left is dedicated to all the metadata about the CSI, including its name, its location and the user who created it, as well as the observation values, both by category and the final value computed through the function implemented in the respective CST. In our example, we can see that the values measured for PM10 and carbon monoxide are respectively 1 and 24.5, thus, the final value computed is 1 as expected, which stands for a good AQI. We can conclude that jogging in such area is safe even for susceptible individuals. The whole platform has been developed using Angular 4 and Django 1.11 and its front-end interface has been designed following the guidelines of Material Design to promote intuitiveness.

## V. Conclusion

In this work we presented a novel IoT paradigm that, contrarily to the vast majority of approaches existing in literature, does not rely on a novel standard or framework oriented to the data gathering, rather, it has indeed the high potential for leveraging resources that are already in place, making use of Collective Awareness Paradigms (CAP). More in detail, we consider particularity the usage of Open Data repositories, being them "reliable", i.e. coming from certified sources, or "unreliable", i.e. crowdsourced through devices belonging to participants who upload freely their data. Furthermore, we also imply Mobile Crowdsensing (MCS) as another data gathering method, for which we implemented a general framework as well as an algorithm that controls sparse and dense data. We deal specifically with many of the research problems that affect such data gathering systems in several past works. Finally, we propose a prototype platform, called SenSquare, designed as a SOA, that allows users to make use of the raw IoT datastreams, collected through collective awareness, composing them in customized services. In particular, we designed a language with customized semantics in order to facilitate inexperienced users to compose services templates and instantiate them into geographical areas according to their needs. We firmly believe that this work opens up a plethora of novel possibilities in research as well as in any entity interested in building IoT applications for the common benefit; in fact, much of the data needed for such applications is already available, although research efforts are needed to make use of it.

## References


[1] Cisco, "Cisco Visual Networking Index: Forecast and Methodology, 2016–2021," Tech. Rep., 2017.

[2] M. Zorzi, A. Gluhak, S. Lange, and A. Bassi, "From today's INTRAnet of things to a future INTERnet of things: A wireless- and mobility-related view," *IEEE Wireless Communications*, vol. 17, no. 6, pp. 44–51, 2010.

[3] C. F. Cargill, "Why standardization efforts fail," *Journal of Electronic Publishing*, vol. 14, no. 1, 2011.

[4] F. Montori, P. P. Jayaraman, A. Yavari, A. Hassani, and D. Georgakopoulos, "The curse of sensing: Survey of techniques and challenges to cope with sparse and dense data in mobile crowd sensing for internet of things," *Pervasive and Mobile Computing*, 2018.

[5] F. Montori, L. Bedogni, and L. Bononi, "A Collaborative Internet of Things architecture for Smart Cities and environmental monitoring," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 592–605, 2018.

[6] R. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: current state and future challenges," *IEEE Communications Magazine*, vol. 49, no. 11, pp. 32–39, 2011.

[7] A. Capponi, C. Fiandrino, B. Kantarci, L. Foschini, D. Kliazovich, and P. Bouvry, "A survey on mobile crowdsensing systems: Challenges, solutions and opportunities," *IEEE Communications Surveys & Tutorials*, 2019.

[8] F. Restuccia, N. Ghosh, S. Bhattacharjee, S. K. Das, and T. Melodia, "Quality of information in mobile crowdsensing: Survey and research challenges," *ACM Transactions on Sensor Networks (TOSN)*, vol. 13, no. 4, p. 34, 2017.

[9] X. Zhang, Z. Yang, W. Sun, Y. Liu, S. Tang, K. Xing, and X. Mao, "Incentives for mobile crowd sensing: A survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 54–67, 2016.

[10] F. Montori, K. Liao, P. P. Jayaraman, L. Bononi, T. Sellis, and D. Georgakopoulos, "Classification and Annotation of Open Internet of Things Datastreams," in *19th International Conference on Web Information Systems Engineering*, 2018.

[11] J.-P. Calbimonte, O. Corcho, Z. Yan, H. Jeung, and K. Aberer, "Deriving semantic sensor metadata from raw measurements," 2012.

[12] F. Montori, L. Bedogni, and L. Bononi, "Distributed data collection control in opportunistic mobile crowdsensing," in *Proceedings of the 3rd Workshop on Experiences with the Design and Implementation of Smart Objects*. ACM, 2017, pp. 19–24.

[13] M. Compton, P. Barnaghi, L. Bermudez, R. GarcíA-Castro, O. Corcho, S. Cox, J. Graybeal, M. Hauswirth, C. Henson, A. Herzog *et al.*, "The ssn ontology of the w3c semantic sensor network incubator group," *Web semantics: science, services and agents on the World Wide Web*, vol. 17, pp. 25–32, 2012.

[14] K. Janowicz, A. Haller, S. J. Cox, D. Le Phuoc, and M. Lefrançois, "Sosa: A lightweight ontology for sensors, observations, samples, and actuators," *Journal of Web Semantics*, 2018.