Crowd-sourced BMS point matching and metadata maintenance with Babel

Conference Paper · March 2016

DOI: 10.1109/PERCOMW.2016.7457102

4 authors, including:

Jonathan Fürst
IT University of Copenhagen
8 PUBLICATIONS **2** CITATIONS
SEE PROFILE

Randy Katz
University of California, Berkeley
386 PUBLICATIONS **35,609** CITATIONS
SEE PROFILE

Crowd-sourced BMS Point Matching and Metadata Maintenance with Babel

Jonathan Fürst*, Kaifei Chen†, Randy H. Katz† and Philippe Bonnet*
*IT University of Copenhagen, †UC Berkeley
jonf@itu.dk, {kaifei, randykatz}@berkeley.edu, phbo@itu.dk

***Abstract*—Cyber-physical applications, deployed on top of Building Management Systems (BMS), promise energy saving and comfort improvement in non-residential buildings. Such applications are so far mainly deployed as research prototypes. The main roadblock to widespread adoption is the low quality of BMS metadata. There is indeed a mismatch between (i) the anecdotal nature of metadata for legacy BMS – they are usually initialized when the BMS is commissioned and later neglected–, and (ii) the imperious need for consistent and up-to-date metadata for supporting building analytics or personalized control systems. Such applications access sensors and actuators through BMS metadata in form of point labels. The naming of labels is however often inconsistent and incomplete. To tackle this problem, we introduce Babel, a crowd-sourced approach to the creation and maintenance of BMS metadata. In our system, occupants provide physical and digital input in form of actuations (e.g., the turning on/off a light) and readings (e.g., reading room temperature of a thermostat) to Babel. Babel then matches this input to digital points in the BMS based on value equality. We have implemented a prototype of our system in a non-residential building. While our approach can not solve all metadata problems, we show that it is able to match end-user relevant points in a fast and precise manner.**

I. Introduction

Non-residential buildings are a prime platform for novel cyber-physical applications that can reduce energy consumption and improve occupant comfort. Reducing energy consumption in non-residential buildings is an important goal, considering that these buildings account for ca. 19% of primary energy consumption in the U.S. [1]. Occupant comfort is likewise important because we spend more than 90% of our time inside buildings [2].

Around half of non-residential buildings are already instrumented with a Building Management System (BMS). A BMS is a typically tightly coupled digital control and sensing system that performs automation and management tasks for a particular building (HVAC, lighting etc.) [1].

Lately, BMS have caught the interest of the Sensor Network community. Previous groundwork has made BMS sensors and actuators available to cyber-physical applications by abstracting them to a common interface (e.g., [3]). Early cyber-physical applications on top of BMS show great potential: Erickson and Cerpa develop a system that allows users to directly communicate their thermal preferences to the BMS, leading to energy savings of 10% and increased personal comfort [4]. Narayanaswamy et al. develop a system that optimizes energy usage and comfort by detecting anomalies in HVAC systems [5].

However, such cyber-physical applications are not yet deployed beyond building-specific research experiments. The problem is metadata. Traditionally, a BMS is commissioned and its functions are hardcoded to the specific characteristics of a building and needs of a building manager. After commissioning, the BMS becomes a legacy system and metadata is largely irrelevant. Devices get replaced and added on a regular basis while spaces get re-configured and change their use (e.g., a classroom is transformed into a lab). Metadata rarely follows such evolution [6].

BMS metadata is usually restricted to a single label that is set manually for each BMS point (e.g., a single sensor or actuator). Specific commissioning contractors are responsible for defining names for these BMS labels. Common naming practices rely on the use of abbreviations for describing building component keywords (e.g., from the HVAC, Lighting domain) and for specifying the location (building, room names). These encodings are not defined by a well-formed namespace. They are not easily parsable, even for humans. Here are some exemplary problems that we found while analyzing the BMS labels in three of our campus buildings:

- No strict naming scheme within a single building. E.g., `WS86007.RELAY12` and `SDH.PXCM-08SDH.S5-04:ROOM TEMP` represents a light switch and the room temperature of a thermostat at the same location.
- Different labels for the same semantic meaning (`RM` and `ROOM`, `TEMP` and `TMP`, `STPT` and `SP`). E.g., `SDH.CP.RESET.TMP.SP.MIN` and `SDH.CAC-3:ROOMTEMP` use different labels for temperature sensors.
- Incomplete metadata due to the restriction to point labels as data structure. For example `WS86007.RELAY12` completely misses information about the location of the light switch.

Inconsistent BMS metadata has been studied before (see Section II for an overview). What distinguishes our approach from previous work is that (i) we apply crowd-sourcing, relying on the building occupants, (ii) we consider both, actuators and sensors while (iii) we provide points with a mapping to their physical location. Our hypothesis is that much of the physical state of a building can be observed by humans and that building metadata maintenance should be based on human input.

In our system, *Babel*, occupants provide physical and digital input in form of actuations (e.g., the switching of a light)

and readings (e.g., the reading of the room temperature of a thermostat). Babel then matches user input to points in the BMS by comparing point values to user provided input. For example, if the user notifies Babel that the temperature is 67^oF, we are able to reduce the qualifying points to all points with value 67. By further iterative refinement of crowd-sourced data (a user reports another value for the same temperature point), we can reduce the qualifying points eventually to a single match. Our intention is to enable occupants to set up the metadata for their own office space by providing input to Babel. After performing this setup process, they are then able to use a personal comfort application, e.g., on their smartphone. We consider BMS metadata maintenance as a good fit for crowd-sourcing for the following reasons:

- BMS metadata maintenance is parallelizable (there can be several parallel user inputs) and serializable (the order of user inputs does not matter).
- Many application relevant BMS points are visible to humans (e.g., thermostat setpoint, light state) and the user effort to report their state is small.
- It can be partitioned (e.g., metadata maintenance in one office is independent from maintenance in another).
- The introduction of our system can improve building operation and allow personal comfort applications. This helps to incentivize people to participate.

The core contribution of our work is the design and implementation of Babel, a system that allows for a incremental, crowd-sourced construction and maintenance of BMS metadata. We implement and evaluate Babel on an actual non-residential building in the U.S. to show the applicability, performance and accuracy of our system. To our knowledge, this is the first work that applies crowd-sourcing to the problem of inconsistent and incomplete BMS metadata.

The remainder of this paper is structured as follows: First, we present related work. We then follow with our design goals and the design of our system. Finally, we presents our deployment and evaluation in a non-residential building, concluded with an outlook on future work.

II. Related Work

Metadata population has been covered extensively for different content. For text, automated metadata generation has been done in various ways, e.g. through natural language processing [7]. Yang and Lee propose a machine learning approach to automatically generate metadata for the semantic web from the content of webpages. Rodriguez et al. use associate networks to transfer metadata from metadata-rich resources to metadata-poor resources. They evaluate their system using a bibliographic dataset [8].

Semi-automated, crowd-sourced metadata generation is very successful in the so-called folksonomies using e.g., community based tagging, where users tag a typically small fraction of documents [9]. An example is the tagging of YouTube videos. Projects like OpenStreetMap go further by collaboratively creating a free editable map of the world that has become comparable if not superior in quality than geo-data from commercial providers [10]. In the building domain, Rice and Woodman have successfully applied the concept of crowd-sourced construction of world models. They present a system that allows occupants to map the inside of a building [11].

In the sensor network community, Bhattacharya et al. proposes a system where sensor metadata is semi-automatically completed using regular expressions to detect common patterns in metadata descriptors using the expertise input of the building manager [12]. They achieve a correct matching of 70% of data points using relatively few manual sample matches. But they note that many labels have a very low frequency, making it very hard to qualify them automatically. Hong et al. apply spatial clustering to classify relative sensor locations with some initial success for 5 rooms and 15 sensors [13]. Finally, Schumann et al. present an approach for the semi-automated mapping of BMS and Energy Management System (EMS) labels. They propose semantic techniques for computing similarity values between BMS and EMS labels. These similarity values are subsequently used to reduce the number of points a user needs to consider when he/she matches labels manually. However they conclude that their approach only identifies the correct match in 16% of all cases and hence is not fit for automated labeling.

To solve the metadata problem in the first place, several long-term efforts to standardize naming exist. Most notable the ISO Industry Foundation Classes (IFC) [15], the open source initiative Project Haystack [16] and ISO 16484-3:2005 [17]. These standards are commonly not adhered to by building constructors and commissioners. Furthermore, a study recently came to the conclusion that none of these metadata schemes fully captures to model a building's sensors and actuators and their relationships [18].

III. Design Goals

We aim to solve the problem of inconsistent and incomplete BMS metadata through crowd-sourcing. Our design goals are the following:

- **Global namespace.** A global metadata namespace enables portable applications across different buildings. Our system should therefore map the current, loose namespace of a building to a global one.
- **Limited user involvement.** To achieve a wide adoption of a crowd-sourced system, user involvement should be limited. Our system should ideally rely on a localization system, so the location of the user can be automatically selected and she does not need to manually input it. Further, Babel should allow for the introduction of a reward system in terms of a salary bonus or internal competition. It should be possible to give users an immediate incentive by e.g., enabling a personal comfort application for them after points have been matched for their office.
- **Backward compatibility.** Buildings have long life-cycles of 50-70 years. This means that our system needs to be able to work with existing BMS and their characteristics (e.g., slow and unreliable data-polling). More recent buildings contain mostly one of the wider used,

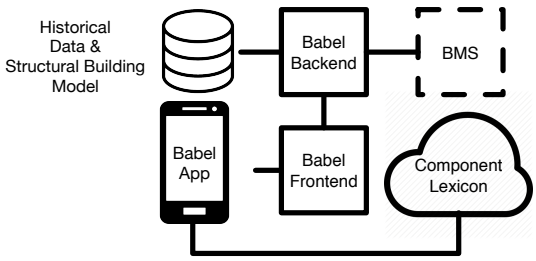

Fig. 1. Babel Architecture

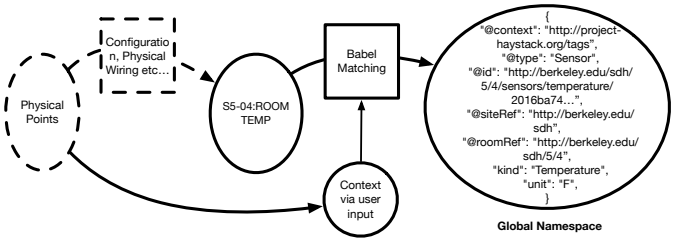

Fig. 2. Mapping the loose BMS Namespace to a Global Namespace

standardized protocols like BACnet, KNX or LONWorks. Many open source and research projects have made it possible to integrate these (e.g., [3]). Our system should build on top of this work. Due to the slow data polling rate of BMS, our system should rely on asynchronous requests where possible.

- **Robustness.** Resulting matches and metadata need to be robust in regards to the human factor. We generally trust the users of our system. We assume that a user will not sabotage our system by providing wrong input. Users will be affiliated to the building as employees or students, which makes this assumption reasonable. Further, in a real deployment, countermeasures against faulty input can be the requirement of *n* correct matches for a point instead of *1* and to distinct between trustworthy and less trustworthy users. However, our system should handle small imperfection and imprecisions in the user input. For example, a user might report 64^oF, while the actual value is 64.3^oF, or she might report an observation with some delay to Babel.
- **Dynamicity.** To achieve dynamic metadata, our system should be able to verify its metadata at defined time-intervals. The time-interval depends on the requirement of the specific building.

In the following we describe our system design and implementation based on these design goals.

IV. System Design

Figure 1 depicts the general architecture of Babel. A smartphone application accesses a Web service in the cloud that contains a lexicon for different device types and points (e.g., a light on/off switch, a thermostat temperature point). The same lexicon is used across all buildings to enforce a global namespace. The smartphone application can further access the local Babel service. It provides (i) the specific, structural model of the building, adhering to a global naming scheme and (ii) an entry point for users to report new values from the physical world. When a user reports a value, it is compared against the current values in the BMS in order to find a match. The matching state is stored in a local database.

A. Metadata Model

Our system requires a data model that is able to represent the different BMS points and respective locations, while allowing to integrate elements outside of the building domain. A cyber-physical application might for instance retrieve local weather information to adjust the HVAC setpoints.

We follow Curry et al. [19] and apply the Linked Data approach to the building domain, solving interoperability issues not only inside that domain, but also with other systems in general. The Resource Description Framework (RDF) is the standard model of the W3C for Linked Data exchange on the Web [20]. RDF has several standardized serializations formats like XML or JSON. In Babel we use JSON-LD (http://www.w3.org/TR/json-ld/) to model the structure of a building and its different components as Linked Data. We principally follow the naming scheme from Project Haystack, which encompasses many points from the BMS domain (e.g., temperature, humidity and CO2 sensors and thermostat setpoints). Points that are not defined in Haystack are included by linking to other naming schemes (e.g., https://schema.org). Figure 2 shows by means of an example how the loose, existing namespace of BMS points is mapped to a global namespace by relying on user provided context. Sensors and actuators make up a majority of BMS points. They are physically wired to a controller, which connects to the BMS backend. Physical sensor and actuator states are then perceived by occupants and serve as parameter for Babel's point matching process. Matches are thus constructed incrementally over time and change as the building changes. This might be the case when the physical points change because part of the HVAC system gets replaced. In the example, a temperature sensor is mapped to a global namespace using the Haystack scheme.

B. Namespace Mapping and Point Matching

To match a single BMS point, we distinguish it from the set of all points through iterative refinement and enrich it with metadata. This is achieved by the simplified matching process seen in Figure 3. We need to eliminate other, unpredictable datapoint changes that might happen during an unfinished matching process. When receiving a new user input, Babel first reduces the points to consider for this request. It removes already successful matches and points whose "BACnet type" value does not fit the type of the point that should be matched.[1] Then it queries the remaining points and compares their value with the user provided value. A substantially reduced list of points is the result. This process is repeated when user input

[1]The BACnet standard defines 54 point types (e.g., Binary Input, Binary Output, Analog Input etc.) [21].

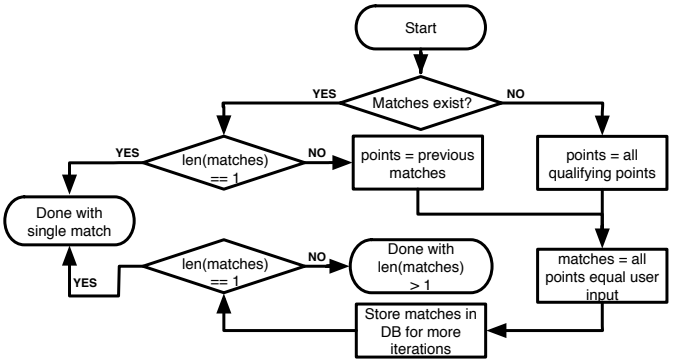

Fig. 3. Babel Matching Process to Distinguish a Single Point.

for the same point is provided again until a single point is left. The point is enriched with metadata in the form of (i) the user selected location and (ii) information from the global component lexicon.

C. Implementation and Deployment

We use one of our campus buildings for implementation and deployment. The 141,000 s.f. 7-story facility contains mainly research spaces and classrooms. The building's BMS is connected to a PC which runs sMAP [3]. We implement a driver for sMAP that runs locally. Our driver uses pyBACnet to query the BMS and spawns a Web service for communicating with the main Babel component.

The main component (written in Go) offers a REST interface for smartphone clients. Most times the driver is idle. When Babel receives user input by a client, it dynamically creates a list of BACnet points that come into question for the user's input and fires up our driver with these points. This reduced list of points is based on BACnet type values and previously matched points. If the point is already part of an ongoing matching process, we continue the process with the already narrowed down points. We implement an Android smartphone client for Babel that connects to a structural model in the form of rooms and a lexicon of point descriptors following the Project Haystack naming convention and modeled in JSON-LD. A user of our app is able to select her location in the building and the type of point she wants to report (e.g., a light switch). She then only inputs the observed state (e.g., on).

V. Experimental Results

We evaluate our system with various micro- and macro-benchmarks. In the following, we present our experimental evaluation on our deployment building. The building is operated by Siemens Apogee, a proprietary BMS by Siemens. It provides a virtual BACnet interface that we interact with.

We scan for BACnet points on the building using pyBACnet. Our scan result shows that the building contains 7053 points. By observing the point names manually, we find that 365 of these points are part of the lighting system, while 6688 belong to HVAC. Analog Output and Binary Output points make for 87% of all points.

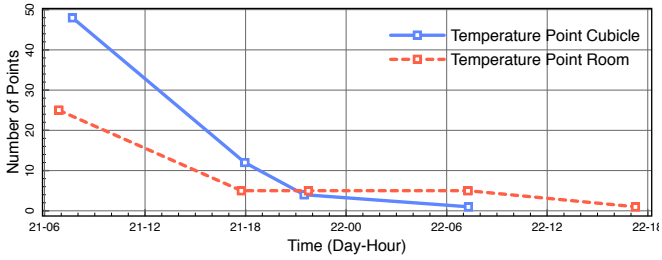

Fig. 4. Matching Progress for Two Thermostat Temperatures

We then sample all BMS points at different times of the day (9am, 1pm and 10pm) to be able to quantify how often the same value is shared among points. When many points have the same value, then a distinction based on the change of that value might require substantial more steps. Our experiments show that a value is shared on average among 4.8 other points. Few values make out the majority for all points (1 and 0 make up 42% of all points). The reason is that our system contains relatively many binary switched lights (365) and that HVAC systems use 1 and 0 to specify their heating and cooling mode.

A. Matching Process

We perform macro benchmarks for the point matching process by physically visiting several thermostats and light switches to perform user input to Babel over the period of one week. Figure 4 shows the result of this process for two thermostat temperature points. The points that need to be considered by Babel reduce over time, dependent on user input and the "grade of singularity" of that value. In depicted case, we are able to reduce the possible points to 48 and 25 with the first user input. In consecutive iterations, we only consider the remaining points. As can be seen for the room temperature (dashed line), the matching process might not be able to reduce the points if either (i) all other possible points have the same value or (ii) some of the points in the BMS do not respond to reading requests.

Figure 5 shows the same experiment for a point that can be directly influenced by users: a binary light switch. There are 365 such points in our deployment building. In contrast to the matching of the temperature setpoint, we can not reduce the number of points as significantly in the first iteration. But then we can observe the strength of our human-in-the-loop approach. By physically switching a light and reporting the new state we could always reduce to a single point during our experiments (100 tries for 20 different points).

To quantify the performance of our system at a larger scale, we have used historical data to simulate an ongoing matching process. The measurements have a 10s granularity. The dataset contains 463 different points. We use this data to perform a point matching in 10 minute time intervals. This means, that we assume that for all unmatched points, every 10 minutes, a user inputs her observations into our system. Figure 6 shows how the distribution of matched points increases over time. The result is a match of all points after 12

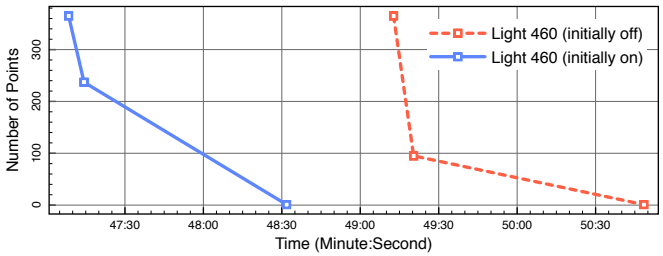

Fig. 5. Matching Progress for Two Light Switches

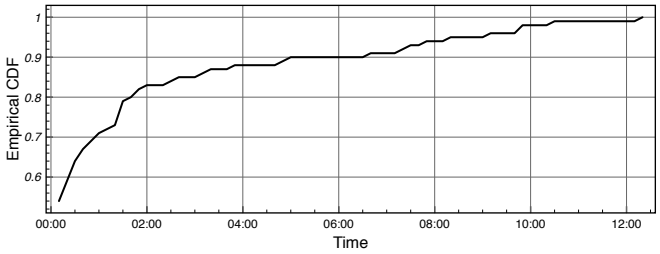

Fig. 6. Cumulative Distribution (CDF) of Successful Matches

hours. One must note that this experiment does not fully model a real deployment. First, this experiment does not contain any user initiated actuation (e.g., light switch, thermostat setpoint changes etc.) that will accelerate the matching process and second, we perform a matching every 10 minutes, which is not realistic for a real deployment. However, the results show that our approach leads to a quick point mapping when users participate sufficiently.

B. End-to-end Latency

We aim for an interactive system, where users get timely feedback on the completion of a matching process. First, low end-to-end latency can enable gamification as user incentive. Second, it enables users to engage in a sequence of rapid interactions (e.g., for lights). Letting users manually change the value of a point through a sequence of interactions with Babel can speed up the BMS metadata matching process.

The experienced end-to-end latency generally correlates with the number of points that need to be queried. Querying all 365 light points results in 5.9s of delay. The time reduces to 4.8 and 3.4s for 237 and 95 points respectively. We found the latency of the BMS the main dominating component for our system and perform therefore micro-benchmarks on it (see Figure 7). We are able to read a bulk of 3224 points in 17.3s (10 tries). The query time does not always correlate to the number of queried points. During our experiments, we found that the query time heavily depends on the physical BMS controller that connects the points. Sometimes querying more points takes less time. E.g., the controller that polls the 2100 points is significantly faster than the controller polling the 1100 points. Such unpredictability makes it challenging to implement an interactive system on top of a BMS.

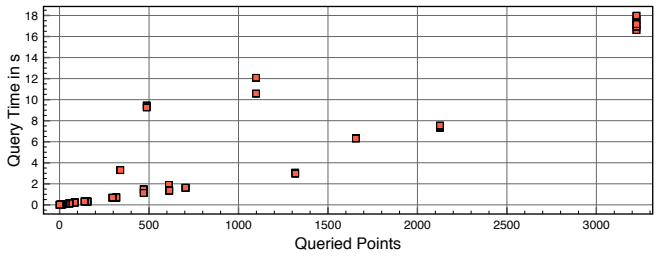

Fig. 7. BMS Query Latency

C. Accuracy

Accuracy is crucial for cyber-physical applications and to overcome reservations by a building manager if a new, more user centric system is installed. If we assume that the state of physical readings by users does always correspond to the digital state in the BMS, then by universal instantiation, this must also be true for a single value that remains at the end:

$$\forall x\, P(x) \Rightarrow P(a/x) \tag{1}$$

This results in a 100% accuracy in an ideal system. However, in a real implementation, we need to consider (i) accidental or intentional wrong user input and (ii) that digital and physical states are not perfectly aligned in time and value dimension. Both can be dealt by building trust through multiple matches by several users. We leave extensive experiments with real occupants for future work, but we experienced that the physical state (e.g., of a light switch or temperature value) is not always equal to the value in the BMS. A thermostat was only able to display temperature as integer, while the value in the BMS was a two decimal floating point number. As a result, we convert values to integers, sacrificing some variance. We also found that physical artifacts do not always correspond one-to-one to BMS points. For instance we found a light switch that corresponded to two points in the BMS. Another problem occurs if BACnet points do not respond to reading requests. If this happens, we might remove the correct point from our list of possibilities. To avoid that, we keep points that do not respond in our list of possible matches.

D. Lessons Learned

The insight we gained over the course of our design and deployment towards the feasibility and performance of our approach is threefold. First, using a BMS as basis for user initiated interaction is feasible as our evaluation shows. However, we often face technological barriers. The BACnet read rate is limited. Considering that we are connecting to a virtual BACnet interface, we suspect a native BACnet to be even slower. Second, our approach has limitations when we deal with points that can not be observed directly by humans. The matching of internal setpoints (e.g., for the control flow of the HVAC system) is out of scope in our work, as these points are not directly relevant for many applications. Third, our matching process is considerably slowed down by a loss of variance due to different encodings of point values (e.g.,

on a thermostat display and on the BMS). We currently deal with this problem by using the lowest common denominator (e.g., convert to integer).

VI. Conclusion and Future Work

In this work, we developed and implemented the concept of crowd-sourced metadata maintenance through point matching for non-residential buildings. Inconsistent metadata in the building space is a problem that hinders the further development of cyber-physical applications that are portable across buildings. Inconsistent metadata is also a problem for traditional building operations. The introduction of a ESM is often expensive and requires manual work [14]. The traditional building commissioning is transforming into a continuous and iterative process that requires consistent metadata to stay cost-effective [22].

Our approach of introducing a human-link between the physical world and the digital point of a BMS works well in the experiments performed in our building. The current implementation is limited by the artifacts an occupant can observe, solving the most relevant metadata problems with regard to end-user applications. In addition, traditional appliances in residential homes are steadily replaced by smart appliances. It is inevitable that these appliances will reach the non-residential building market during the next years. This plethora of new devices will be much more interactive for humans and align itself well with our idea. Ultimately, we envision Babel as a general approach to metadata maintenance in the context of IoT.

Looking forward, we see several directions for future work. The obvious next step is to deploy Babel with actual occupants of a building to investigate to which extend our approach is an acceptable effort. Further, gamification is an ideal candidate for Babel. Ingress is a popular game (7 million players) by Niantic Labs (Google) where the goal of players is to conquer geographical areas in the real world by physically visiting these places (https://www.ingress.com). Combining Ingress's approach with metadata maintenance appears to be promising. Lastly, building commissioning is a process that often is only completed once when the building is constructed. A commissioning process that is driven by human input could propagate a continuous commissioning process where occupants keep the physical state of the building in sync with the digital.

Acknowledgment

This work is supported in part by the National Science Foundation under grant CPS-1239552 (SDB).

References

[1] U. E. P. Agency, "Building energy data book," 2011.

[2] U.S. EPA/Office of Air and Radiation, "The inside story: A guide to indoor air quality,," 1988.

[3] S. Dawson-Haggerty, X. Jiang, G. Tolle, J. Ortiz, and D. Culler, "sMAP: a simple measurement and actuation profile for physical information," in *SenSys'10*. ACM, 2010.

[4] V. L. Erickson and A. E. Cerpa, "Thermovote: participatory sensing for efficient building hvac conditioning," in *BuildSys'12*. ACM, 2012.

[5] B. Narayanaswamy, B. Balaji, R. Gupta, and Y. Agarwal, "Data driven investigation of faults in hvac systems with model, cluster and compare," in *BuildSys'14*. ACM, 2014.

[6] T. Haasl and T. Sharp, *A practical guide for commissioning existing buildings*. Oak Ridge National Laboratory, 1999.

[7] H.-C. Yang and C.-H. Lee, "Automatic metadata generation forweb pages using a text mining approach," in *WIRI'05*. IEEE, 2005.

[8] M. Rodriguez, J. Bollen, and H. Sompel, "Automatic metadata generation using associative networks," *TOIS'09*, 2009.

[9] A. Mathes, "Folksonomies-cooperative classification and communication through shared metadata," 2004.

[10] P. Neis, D. Zielstra, and A. Zipf, "The street network evolution of crowdsourced maps: Openstreetmap in germany 2007–2011," *Future Internet*, vol. 4, no. 1, pp. 1–21, 2011.

[11] A. Rice and O. Woodman, "Crowd-sourcing world models with openroommap," in *PERCOM'10*. IEEE, 2010.

[12] A. Bhattacharya, D. Hong, D. Culler, J. Ortiz, K. Whitehouse, and E. Wu, "Automated metadata construction to support portable building applications," in *BuildSys'15*. ACM, 2015.

[13] D. Hong, J. Ortiz, K. Whitehouse, and D. Culler, "Towards Automatic Spatial Verification of Sensor Placement in Buildings," *BuildSys'13*, 2013.

[14] A. Schumann, J. Ploennigs, and B. Gorman, "Towards automating the deployment of energy saving approaches in buildings," in *BuildSys'14*. ACM, 2014.

[15] buildingSMART, "IFC Standard," http://www.buildingsmart-tech.org/specifications/ifc-overview, 2015.

[16] Project Haystack, http://project-haystack.org/, 2015.

[17] ISO, "ISO 16484-3:2005," 2005.

[18] A. Bhattacharya, J. Ploennigs, and D. Culler, "Analyzing metadata schemas for buildings: The good, the bad, and the ugly," in *BuildSys'15*. ACM, 2015.

[19] E. Curry, J. O'Donnell, E. Corry, S. Hasan, M. Keane, and S. O'Riain, "Linking building data in the cloud: Integrating cross-domain building data using linked data," *Advanced Engineering Informatics*, 2013.

[20] W3C, "Resource Description Framework (RDF)," http://www.w3.org/RDF/, 2014.

[21] S. T. Bushby, "BACnet TM: a standard communication infrastructure for intelligent buildings," *Automation in Construction*, vol. 6, no. 5, pp. 529–540, 1997.

[22] E. Mills, *The cost-effectiveness of commercial-buildings commissioning: A meta-analysis of energy and non-energy impacts in existing buildings and new construction in the United States*. Lawrence Berkeley National Laboratory, 2004.

[23] J. Fürst, K. Chen, R. H. Katz, and P. Bonnet, "Demo abstract: Human-in-the-loop bms point matching and metadata labeling with babel," in *BuildSys'15*. ACM, 2015.