Poster Abstract: Automated Metadata Transformation for A-Priori Deployed Sensor Networks

Arka Bhattacharya [*], David Culler
Dept. of Electrical Engineering and Computer Sciences
University of California, Berkeley
arka,culler@eecs.berkeley.edu

Dezhi Hong, Kamin Whitehouse
Dept. of Computer Science
University of Virginia
hong,whitehouse@virginia.edu

Jorge Ortiz
IBM TJ Watson Research Center
jjortiz@us.ibm.com

Abstract

Sensor network research has facilitated advancements in various domains, such as industrial monitoring, environmental sensing, etc., and research challenges have shifted from creating infrastructure to utilizing it. Extracting meaningful information from sensor data, or control applications using the data, depends on the metadata available to interpret it, whether provided by novel networks or legacy instrumentation. Commercial buildings provide a valuable setting for investigating automated metadata acquisition and augmentation, as they typically comprise large sensor networks, but have limited, obscure metadata that are often meaningful only to the facility managers. Moreover, this primitive metadata is imprecise and varies across vendors and deployments.

This state-of-the-art is a fundamental barrier to scaling analytics or intelligent control across the building stock, as even the basic steps involve labor intensive manual efforts by highly trained consultants. Writing building applications on its sensor network remains largely intractable as it involves extensive help from an expert in each building's design and operation to identify the sensors of interest and create the associated metadata. This process is repeated for each application development in a particular building, and across different buildings. This results in customized building-specific application queries which are not portable or scalable across buildings.

We present a synthesis technique that learns how to transform a building's primitive sensor metadata to a common namespace by using a small number of examples from an expert, such as the building manager. Once the transformation rules are learned for one building, it can be applied across buildings with a similar primitive metadata structure. This common and understandable namespace captures the *semantic* relationship between sensors, enabling analytics applications that do not require apriori building-specific knowledge.

Initial results show that learning the rules to transform 70% of the primitive metadata of two buildings (with completely different metadata structure), comprising 1600 and 2600 sensors, into a common namespace ([1]) took only 21 and 27 examples respectively(Figure 1c). The learned rules were able to transform similar primitive metadata in other buildings as well(Figure 1d), enabling writing of portable applications across these buildings. The techniques developed here may be applicable to the other large legacy sensor networks, such as industrial processing, or urban monitoring.

[*]Primary Author.

SenSys'14, November 3–5, 2014, Memphis, TN, USA.
ACM 978-1-4503-3143-2/14/11.
http://dx.doi.org/10.1145/2668332.2668369

1 Brief Technique Overview

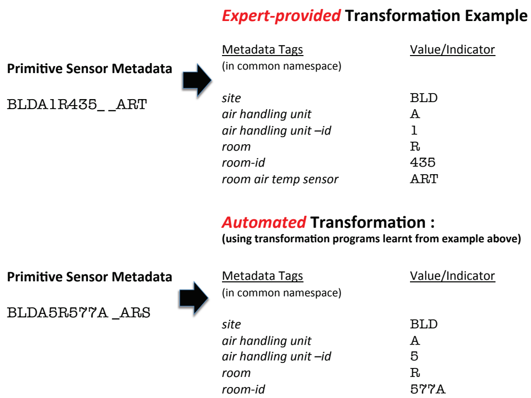
Our technique automatically synthesizes regular expression programs that transform primitive sensor metadata into a common desired namespace. Learning from expert-provided examples has two advantages over manually generating regular expression programs — (a) the experts, often facility managers or maintenance professionals, are not well-equipped to construct the correct regular expression programs themselves; and (b) inconsistencies in the metadata structure and the large number of programs required makes manual generation expensive and error-prone.

Figure 1a shows an instance of an *expert-provided example*, where the expert provides the metadata tags(from the common namespace [1]) that can be derived from the primitive metadata. For example, the presence of the substring `ART` indicates that the sensor is a *room air temp sensor*, and `435` is the value for the *room-id*.

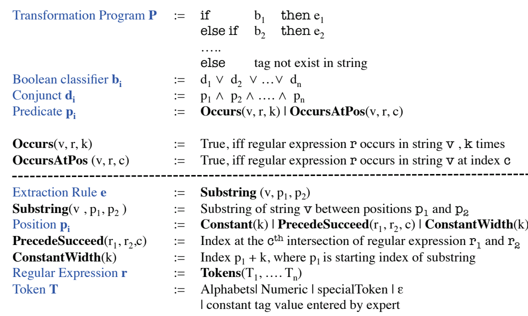
Once the expert has provided an example, we use the language shown in Figure 1b to synthesize for each tag (a) the set of all transformation programs (regular expressions) that extract its corresponding substring, and (b) a set of *If .. Then .. Else* boolean conditions to indicate whether a particular metadata tag from the common namespace should be present in a sensor's transformed metadata.

Our technique then applies the transformation programs on remaining sensors. If there exists a sensor whose primitive metadata could not be fully transformed by the current set of learned programs, it is presented to the expert for manual transformation[1]. As the expert provides more ex-

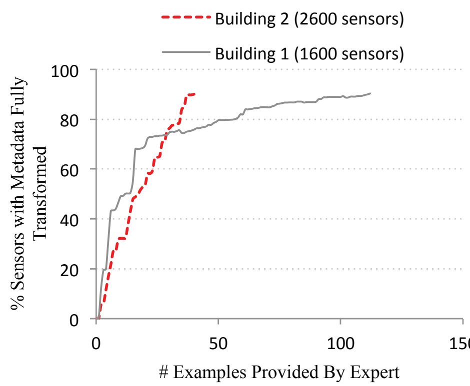
[1]The exact criteria for the choosing the next example to be presented to



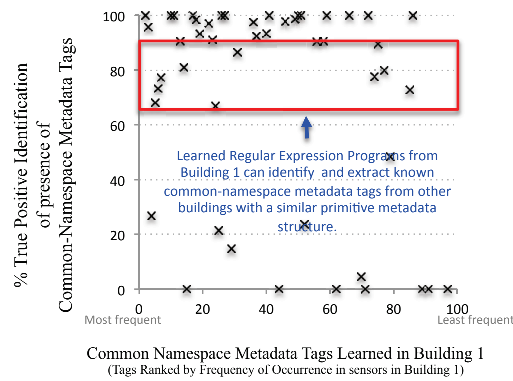
(a) Instance of expert-provided metadata transformation example and automated metadata transformation from the rules learned from that example. Note that the expert has not yet given an example showing how to transform the primitive metadata string `ARS` to a metadata tag in the common namespace.



(b) Language for learning metadata transformation (Similar to [3])



(c) The number of examples required to fully transform primitive sensor metadata in two buildings to the common namespace. Complete transformation of metadata of 70% of sensors require only 21 examples for Building 1 and 27 examples for Building 2.



(d) Applying Regular expressions programs learned in Building 1 to 55 other buildings whose sensors had a similar primitive metadata structure.

Figure 1: Description and Performance of the synthesis technique to learn metadata transformation rules. For more details refer to [2]

amples, we prune the set of synthesized transformation programs such that they are consistent with every example. The evaluation of this technique is shown in Figure 1c and 1d.

2 Applications on Transformed Namespace

The transformation of primitive metadata into the common namespace enables the development of applications, such as rogue zone detection, which are portable across the building stock. A zone (or room) is *rogue* if its air temperature is constantly above or below their required setpoints, i.e it requires constant cooling or constant heating. Such zones constrain air settings and constrain efficiency improvements. To find the rogue zones in a building, one needs to simply search for sensors having the **room air temp** tag, and for each such sensor, find a sensor with the tag **room air temp setpoint** having the same value for **room-id**.

3 Ongoing and Future Work

We are currently working to devise computationally efficient ways of handling large number of examples,evaluating the robustness of our approach on a wider set of buildings. This work was supported in part by NSF grants CPS-0931843 (ActionWebs), CPS-1239552 (SDB) and the DOE OpenBAS grant.

4 References

[1] Project haystack. `http://project-haystack.org/`.

[2] A. Bhattacharya, D. E. Culler, J. Ortiz, D. Hong, and K. Whitehouse. Enabling portable building applications through automated metadata transformation. Technical Report UCB/EECS-2014-159, EECS Department, University of California, Berkeley, Aug 2014.

[3] S. Gulwani. Automating string processing in spreadsheets using input-output examples. In *Proceedings of the 38th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, POPL '11, pages 317–330, New York, NY, USA, 2011. ACM.

the expert for transformation is a focus of ongoing research (see [2]).