IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 10 October 2023, accepted 19 November 2023, date of publication 21 November 2023, date of current version 13 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3335595

RESEARCH ARTICLE

Automated Data Model Generation From Textual Specifications: A Case Study of ECHONET Lite Specification

VAN CU PHAM[1], **(Member, IEEE), NGUYEN THI DIEU LINH**[2], **TUNG LE**[2], **TIEN HUY NGUYEN**[2], **AND YASUO TAN**[1], **(Member, IEEE)**

[1]Center for Digitalization Endeavors, Japan Advanced Institute of Science and Technology, Nomi, Ishikawa 923-1211, Japan
[2]Department of Science and Technology, Hanoi University of Industry, Hanoi 100000, Vietnam

Corresponding author: Van Cu Pham (cupham@jaist.ac.jp)

This work was supported by the ECHONET Consortium through Collaborative Research Efforts.

ABSTRACT ECHONET Lite stands as a leading protocol for smart home appliances in Japan, and the publication of data models plays a critical role in fostering collaboration with other ecosystems, not just for ECHONET Lite but for any protocol or standard. Typically, data models are meticulously crafted by experts through the arduous task of condensing and summarizing extensive specification documents. As an illustration, generating a data model solely for the ECHONET Lite protocol (without incorporating other protocols) can demand thousands of working hours. This paper presents an AI-driven solution aimed at alleviating the burden of laborious, repetitive tasks prone to errors during the creation of data models from ECHONET Lite specifications, automating these processes to save human effort. The proposed solution employs Natural Language Processing techniques to extract key vocabularies from natural language descriptions of the specification, mirroring the approach of experts. Consequently, this solution can generate several data models for the ECHONET Lite protocol in a matter of seconds, all using a standard laptop. The findings indicate that machines are capable of emulating experts in extracting vocabularies, ensuring both syntactic error-free outcomes and consistency in the generated data models. Furthermore, the machine offers rapid, dependable results and enhances the reusability of exported data models across various platforms. The generated data models meet the same requirements as those created by humans. This solution is integrated into an official workflow for generating data models for the ECHONET Lite web API and others.

INDEX TERMS Data models generation, ECHONET device objects, ECHONET lite web API, machine-generated data models.

I. INTRODUCTION

Web Application Programming Interfaces (Web APIs) [1], [2], serve as the fundamental infrastructure for the World Wide Web, cloud systems, mobile applications, and, more recently, the Internet of Things (IoT). They enable the encapsulation of IoT device resources as APIs, allowing them to be accessed through standard Web protocols, as discussed in [3]. In response to this trend, the ECHONET consortium has introduced ECHONET Lite Web API (ELWA) guidelines, necessitating the creation of data models, akin to the *WoT Thing Description* as described in [4], to define available resources and supported interfaces for interacting with EL devices (via the Web).

Until now, the ECHONET Lite (EL) facilitates interoperability among smart 'Home Appliances' and 'Home Gateway' devices manufactured by different vendors within local home networks. This is achieved through the publication of a data model as an international standard (IEC 62394:2022 [5]), known as 'ECHONET Device Objects' (EDO) [6]. The

The associate editor coordinating the review of this manuscript and approving it for publication was Chin-Feng Lai.

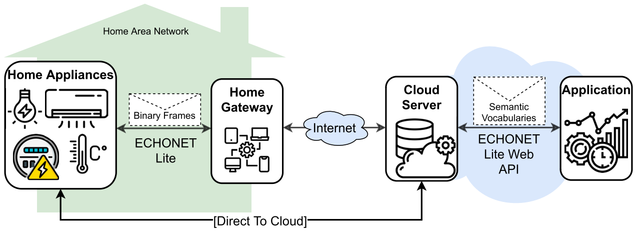

FIGURE 1. ECHONET Lite Ecosystem Reference Architecture. The ECHONET Lite data model is represented using hexadecimal code. The ECHONET Lite Web API, on the other hand, employs a JSON data model with key-value pairs, where each key corresponds to the semantics of the hexadecimal code.

TABLE 1. An example of describing an air conditioner in accordance with the ECHONET Lite specification: The pair *0xAE:0x42* indicates that the air conditioner is currently in Cooling mode.

Package-type commercial air conditioner (Indoor Unit)			
Property name	**EPC**	**Contents of property**	...
		Value range (decimal notation)	...
...	...	...	
Current function (automatic operation mode)	0xAE	This property indicates, when the air conditioner is opening in the "automatic" operation mode, the function ("cooling", ’heating", "dehumidification", "air circulation" , or "other") that is currently being used.	...
		The following values shall be used: Cooling: 0x42 Heating:0x43 Dehumidification: 0x44 Air circulation: 0x45 Other: 0x40	...
...	...	...	...

EDO specification defines the accessible properties of home appliances using hexadecimal codes, an example of the EDO is summarized in Table 1.

In order to drive the implementation and progress of the ELWA, it is essential to turn these hexadecimal codes into a new data model that is user-friendly for both developers and individuals who may not be familiar with the EDO specification. For instance, the pair $0xAE$: $0x42$ should be described as “*currentFunction*”: “*Cooling*”. At present, the development of the data model for the ELWA relies on human experts who apply their expertise to map hexadecimal codes to semantic vocabularies, referencing information from the EDO specification. However, this approach is labor-intensive and vulnerable to human errors, which can encompass syntax mistakes, typographical errors, and discrepancies in definitions.

The automation of data model generation by machines is anticipated to tackle the limitations associated with human involvement. As demonstrated in Table 1, employing a simple rule-based approach enables a machine to effortlessly, consistently, and swiftly map the $0xAE$ to *currentFunction*. While this may appear to have a straightforward solution, the complexity of longer descriptions, often encountered in the EDO descriptions, makes the rules-based approach impractical.

Recent advancements in machine learning (ML), particularly in natural language processing (NLP) [7], have opened new possibilities. It is anticipated that the limitations of the rules-based approach can be mitigated by utilizing NLP techniques to grasp the semantics of the original description and streamline it into a key vocabulary, mirroring the process employed by experts in crafting data models. This research presents the concept of automating the generation of data models for the ELWA from EDO as an initial step, and subsequently extending this approach to create data models for other EL-related data formats, such as the FIWARE smart data model, Web of Things’ Thing Description, and EL-SAREF ontology by simply adding conversion rules, all through this machine-based approach. The primary goals of this study are the following.

- Investigate and implement an NLP solution to analyse semantic of EDO’s decription and turn them into vocabularies of a data model. The primary objective of this solution is to acquire insights from experts on how they effectively remove extraneous words from descriptions, enabling the creation of concise and meaningful vocabularies derived from EDO descriptions.
- Propose a rule-based approach for the automated generation of data models (machine-generated data models) from EDO, with the objective of ensuring compliance with both the syntactical data model (ELWA[1]) and the semantic data model (the SAREF family ontology [8]). The primary goal is to produce data models that exhibit consistency, speed, without syntactical errors, allowing for a comparative assessment against data models crafted by humans.

The rest of this paper is organized as follows. In Section II, concepts related to the ECHONET device objects and current effort in supporting data models for the ECHONET device objects specification are briefly introduced. Section III highlights the overall concept and main building blocks of the proposed solution. Section IV and Section V describes the building blocks of the proposed solution in greater details. Moreover, the design, implementation, and evaluation are also described. Finally, our work is summarized in Section VI.

II. RELATED WORK

As stated in the ITU-T Y.2070 [9], a data model is recommended at the *Device*, *Home Gateway*, and *Management Platform* layer. The data model at each layer has different purposes and characteristics. The EDO supports the *Device* layer. The data model for the ELWA, namely *Device Description*, is located above the *Management Platform* layer. In [10], an ontology-based data model was introduced at the *Home Gateway* layer. The proposed data model supports the ECHONET device objects specification (release K) and ambient assisted living platform [11]. However, those kinds of data models are directly derived from the EDO and all of them have been manually created.

[1] https://echonet.jp/web_api/

A. ECHONET DEVICE OBJECTS

The EDO is a part of the ECHONET Lite [12], a leading protocol for smart homes in Japan [13]. As illustrated in Figure 2, a device object represents a logical device that is classified into seven groups and 117 classes of devices in the latest English specification released in 2020 (Release M). Device objects offer a standardized method to represent device resources and services via a list of *Property* and constraints for each property via hexadecimal codes. Besides the official paper-based document of the EDO, in [14], the author proposed an idea of providing a JSON version of the ECHONET device objects, and the JSON schema that supports the latest version of the device object is available at the *HEMS Interoperability Test Center*.[2]

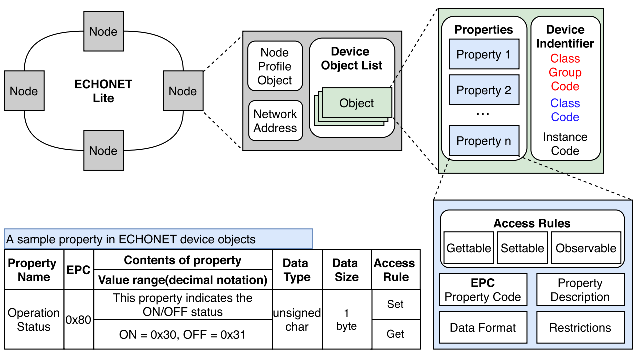

FIGURE 2. The concept of ECHONET device objects.

B. ECHONET DEVICE DESCRIPTION

Since data models and descriptions are extremely important to support the Web API [15], the developments of data models at the *Management Platform* layer are emerging. For example, FIWARE [16] provides several JSON-based smart data models for smart city infrastructure and the developments of new data models for other "smart" infrastructures are actively promoting. The oneM2M ecosystem released ontology-based semantic data model [17], and also smart device templates (syntactic). However, the support of the EDO has not been introduced in those platforms.

The *Device Description* (DD) [18] was created by the *ECHONET Lite Web API Working Group*. A total of 43/117 JSON schemes that describe the EDO have been released, however, only **27** out of **43** schemes are valid where the rests need to be revised and 4 of them require major reworking. As it was created by humans, common mistakes of this data model include spell mistakes, syntactic mistakes (missing fields, misplaced fields, incorrect use of quotation marks), incorrect descriptions of data types.

In a report, it took approximately 3 hours for an experienced person to create JSON schemes for 6 sensor objects with only 9 properties. It is such a time-consuming task to create data models for the ECHONET device objects specification manually. To this end, the automatic generation of data models by machines is desired to handle the drawbacks of humans. Nevertheless, the biggest barrier of using a machine is that there is no rule to map the descriptions of hexadecimal codes from the EDO into vocabularies so far. Therefore, instead of using rule-based approaches, the machine needs to understand the meaning of the descriptions then shorten the descriptions into meaningful vocabularies.

C. NAMING CONVENTION STUDIES

Although experts create vocabularies by shortening the descriptions, it is not easy to create the shortening rules because descriptions are written in natural language. Nevertheless, in recent years, there are NLP techniques that help machines to understand human natural languages.

In [19], authors utilize the n-gram language model to improve the consistency of naming conventions for programmers by learning the naming style of humans and suggesting revisions when an inconsistent naming convention is detected. However, it requires a large dataset to obtain the word and phrases' probabilities. Besides, the traditional language model by n-gram occurrences easily gets the noise from stop-words and redundant information in the dataset. Even the n-gram language model is a non-contextual consideration that eliminates the word sense and its context. In our model, we take advantage of the pre-trained language model, BERT. This language model is regarded as contextual text understanding in a huge dataset (like Wikipedia). Therefore, it is sufficient to utilize the sense, position, and relationship of words and sentences with their contexts.

In [20], the authors propose a solution based on BERT for code comments generation. This work also shows that BERT is more effective and promising than the traditional language models. The proposed solution utilizes the BERT architecture to learn the sequence of keywords and design the BERT-C for code embedding. The code comments generated by the probability distribution at the *softmax* layers are dependent on the previous predicted-string. However, this supervised-learning model requires around 100 thousand samples to obtain the own BERT for this specific task. In our goal, we expect to design an unsupervised-learning model with less effort to extract the important keywords instead of generating the new ones without gold data. With the strength of pre-trained BERT, our model is not only compact but also powerful enough to determine the essential information in the descriptions to create the shorter ones.

In [21], authors propose a solution to debug the naming convention of a software program. This work derivates the specific rules by the observation and statistic analysis in the corpora. The limit of statistic models is noise's sensitivity and data requirement. With less effort and data, our model is also promising to determine which attributes are important and popular through their context. Besides, this work is not applicable to generate vocabularies for the EDO because the

[2]http://sh-center.org/files/156

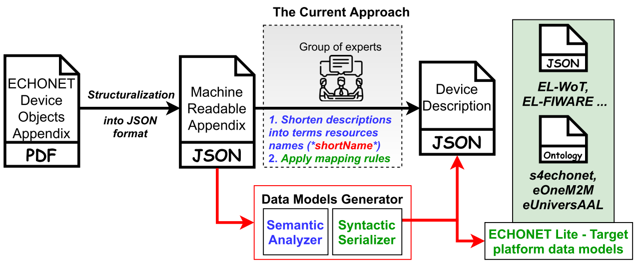

FIGURE 3. The concept of automated generation of data models from ECHONET device objects specification.

proposed solution is limited in the number of return types meanwhile there is no limitation for the naming of the DD.

III. AUTOMATED GENERATION OF DATA MODELS: THE CONCEPT

The overall concept of the proposed solution, namely *eModelGen* is illustrated as in Figure 3. The input of the whole process is the EDO. The JSON version of the specification is available as an open document so that the process to digitalize the input data is unnecessary. The structure of the provided input document is defined as followings:

- Metadata indicates overall information of the documents such as version information, release number, etc.
- A list of supported data type definitions.
- A list of device objects. A device object provides (i) the object code (EOJ), (ii) the name of the object in English and Japanese, and (iii) a list of properties supported by the object.
- A property provides (i) the property code (EPC), (ii) the name of the property in English and Japanese, (iii) access rules of the property, and (iv) data type of possible values of the property.

Since the document is created to support developers to interact with hardware devices, the *object code* and *property code* are acceptable. However, for the Web APIs development, the *object name* and *property name* are fundamental. Therefore, a process to extract semantics of the *object name* and *property name* descriptions to create **vocabularies** is inevitable. Currently, this process is manually achieved by experts and the *eModelGen* has a *Semantic Analyzer* (in Section. IV) to mimic this process.

Even though the semantic vocabularies are the same for different models to enhance consistency and reusability, each data model has different syntactic formats as well as data type definitions. Therefore, for a target data model, a target model *Generator* (in Section. V) is required.

IV. SEMANTIC ANALYZER

In the development of this building block, we realize that a concise representation of object and property names as vocabularies can reduce time and effort to create data models. It comes from a ton of redundancy from unnecessary information in the description string. In recent systems, a formation of property and object names is carried out by humans and it is too difficult and burdensome to deploy in the huge of samples in the practical environments. Therefore, it motivates and encourages us to build an unsupervised-learning and automatic system to extract short and meaningful vocabularies from the long descriptions of the specification. Based on the requirement of this area, we prefer concise patterns to the long ones. It means that the shorter the extracted phrases are, the better results we get. In other sense, the representation needs to maintain the semantic of these original descriptions.

To build the NLP model, features of sentences and words (from description sentences) are extracted by Key-phrase Feature Extraction in Section IV-A. Then, we apply our Description Extraction algorithm to obtain the critical components from the original input in Section IV-B. Next, to prove the effectiveness of our model, the result of our model as well as the comparison with the manual samples are represented thoroughly in Section IV-C. Besides, we also present some samples and discussion in Section IV-C4.

A. KEY-PHRASE FEATURE EXTRACTION

The most important goal in the key-phrase extraction is represent a property name into the meaningful space. Each value in the representation reflects the linguistic features of object and property names. In Natural Language Processing (NLP), there are two main approaches to extract the features. While rule-based approach needs much more efforts from the expert in the specific domain, neural-based models is ideal to learn the features automatically. Through the development of neural-based approaches, BERT [22] is one of the famous and popular language models based on Transformer architecture [23]. This model is trained on the huge datasets to learn the linguistic relationship and meaning of words and sentences via their context.

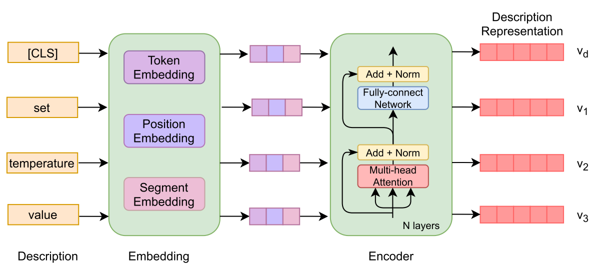

FIGURE 4. Sentence feature extraction model.

Particularly, BERT is basically a stack of Transformer [23] blocks where each word is interacted with the other via multi-head attention mechanism. The representation of words is built on the context of sentence. Through multi-head attention, the meaning and relationship of each word in the sentence is emphasized gradually during training process. Besides, this mechanism allows one word to have multiple sense, which is unavailable in the previous approaches.

Moreover, the language model, BERT, is effective to digest lexical and semantic word features through bi-directional learning mechanism. While the content of word depends on the left context in previous words, BERT allows the word representation to observe both right and left context.

With the aid of transfer learning, our feature extraction utilizes the pre-trained weights of BERT model to derive the linguistic features of object and property names. The detail of our feature extraction is presented in Figure 4. In this module, the input is object and property description. In the pre-processing step, the special character, $[CLS]$, is added at starting position of the input. In most BERT-based approaches, $[CLS]$ token is considered to have an ability for generating the meaning of sentence. In our feature extraction, we utilize the output of last layer in BERT model to represent the sentence and its words.

B. DESCRIPTION EXTRACTION

The goal of description extraction is to condense the object and property names by reducing the redandant words in the original input. It requires the scoring function to determine an important score of each word. This function has to reflect the difference in meaning between description and words. In our Description Extraction, we propose to utilize the cosine similarity to measure the spatial distance between sentence and word representation. Specifically, after obtaining the features of the description in Equation 1, the cosine similarity between sentence and each word is calculated in Equation 2

$$v_d, \{v_i\}_1^{|d|} = BERT(d) \in \mathbf{R^N} \tag{1}$$

$$s_i = sim(v_i, v_d) = \frac{v_i v_d}{||v_i|| \, ||v_d||} = \frac{\sum_n^N v_i^n v_d^n}{\sqrt{\sum_n^N v_i^n}\sqrt{\sum_n^N v_d^n}} \tag{2}$$

where d is the descriptions whose vector is v_d and the representation of i-th word in d is v_i.

Based on the distance between the sentence and its words representation, the concise description is formed by all words in the cluster of the original input as Figure 5. Obviously, the size of this cluster is the fundamental question in this case. However, the number of words in description's cluster is based on the acceptance rate from users and domains, which has been ever mentioned in previous works. Therefore, we propose to utilize the Algorithm 1 to extract the important words under the threshold b.

Particularly, to expand the selected cluster c, we select the words that are higher cosine similarity s_i than lower ones. However, we expand the set c through the threshold b with the cost function $f(.)$. Since the standard cost function has not been defined in this step, we propose a ratio that corresponds to the percentage of selected words and the original description as in Equation 3. The reason for our proposal comes from our observation of this task. The shorter vocabularies is, the better result we obtain.

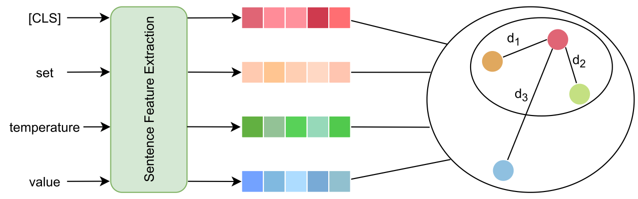

FIGURE 5. **Description extraction by clustering words into the sentence' cluster via cosine similarity.**

Algorithm 1 Description Extractor: Select Important Words From Description **d** Under The Budget **b**

Input: description d, threshold b
Result: List of words
$v_d, \{v_i\}_1^{|d|}$:= BERT(d);
while $i < |d|$ **do**
  $s_i :=$ sim(v_i, v_d);
end
$\bar{W_i} := sorted(\{w_i, s_i\}_{i=1}^{|d|})$;
$c := \emptyset$;
while $((f(c) < b))$ **do**
  $c := c \cup \bar{s_k}$;
end
out := reorder(c) ;

$$f(c) = 100\frac{|c|}{|d|} \tag{3}$$

Finally, we assume that the order of selected word is similar to its position in the original description, which is controlled by the $reorder(.)$ function.

C. EXPERIMENTS

1) DATASET AND SETTINGS

Dataset for the experiment is extracted from the *JSON Device Description* provided by the ECHONET consortium. From the *JSON Device Description*, samples are extracted as followings:

- Device name:
  - Input: The *descriptions.en* field of a JSON schema which is a sentence to describe the object.
  - Output: The *deviceType* field of the JSON schema which is a abbreviation manually created by humans.
- Property name:
  - Input: The *property.descriptions.en* field of a JSON schema which is a sentence to describe the property.
  - Output: The *propertyName* field of the JSON schema which is a abbreviation manually created by humans.

The summary of the dataset is as in Table 2.

In experiments, we take advantage of the pre-trained BERT with the *'bert-base-uncased'* mode instead of deploying from

TABLE 2. The detail of dataset.

	Value
Number of Samples	470
Avg Number of Word: Input	4.69
Avg Number of Word: Output	3.22

TABLE 3. The human evaluation among human-generated, TF-IDF and our model's outputs.

	% Selelected by Human
Human-generated	23
TF-IDF	25
Our model	**52**

the scratch. In addition, the threshold b is in the range from 5 to 100 with the step of 5 which reflects the remaining information of description in the extraction phase.

2) EVALUATION METRICS AND RESULTS

As the previous parts, the goal of our work is to reduce the human effort on generating the vocabularies. However, the machine-generated output should be close to the human-created ones and the input in meaning. Specifically, the quality of a generated sample is evaluated by (i) the human evaluation in practice; (ii) the word deviation and length between machine-generated samples and human ones.

Firstly, we employ human to evaluate the efficiency of our model-generated outputs compared against the human-generated outputs and the TFIDF-generated outputs where TF-IDF [24] is utilized to represent sentences and words in term of word frequency. Particularly, the respondents is provided three description names of corresponding to (i) human-generated, (ii) our model-generated, and (iii) the TF-IDF generated. Without any information of approach's name, the participants are required to choose the best suitable output for representing the given input. In this evaluation, the acceptance score of humans can reflect the feasibility of machine learning in description extraction to reduce the manual cost in previous works. According to the results of 420 samples showed in Table 3, the proposed approach with the threshold $b = 80$ significantly outperforms the others. As our model efficiently identifies essential words, the outputs are more meaningful and distinguishable. In Section IV-C4, we analyze some cases to show the advantages and drawbacks of these approaches.

Secondly, as we mentioned above, we prefer the shorter vocabularies than the longer one. Therefore, we use two popular metrics in word-based distance including **Edit Distance** (ED) and **Word Error Rate** (WER) to get the deviations of our machine-generated outputs and the human-created outputs from original inputs. It means that we compare the metrics of human-vs-inputs and machine-vs-inputs together. In this evaluation, the smaller deviation score shows better results. The result of this evaluation is presented in Table 4 at the threshold $b = 80$.

TABLE 4. The comparison between human-created and machine-generated outputs at threshold $b = 80$.

	Edit Distance	Word Error Rate
Human-generated	16.02	0.47
TF-IDF	7.51	0.39
Our model	**7.43**	**0.35**

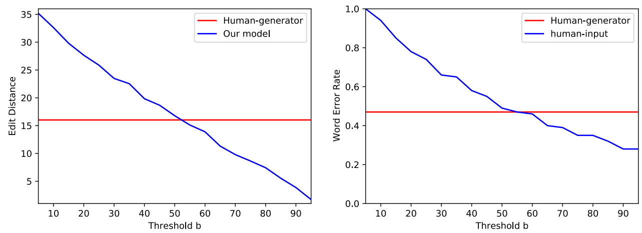

FIGURE 6. The effort of threshold into the generation phase.

WER and **ED** score put the concentration on the modification in the mophological consideration. The higher score is equivalent to the more differences between the input and generated output. In this evaluation, our model with the approximate information ratio outperforms the human version. It reflects that our generated samples need less effort to extract from the description input.

3) ABLATION STUDIES

In this part, we also present the effect of threshold b in our generation phase in Figure 6.

We observe that the **WER** and **ED** has the same trend. When the information ratio - threshold is large, the selected set is close to the input. At the threshold of $50-60$, our model achieves the same quality against the human-created samples.

4) DISCUSSION

In this section, three typical cases include **Good**, **Bad**, and **Fair** of machine-generated property names are discussed. A good case is as followings:

- Input: *vegetable compartment temperature setting*
- Human: *vegetableTemperature*
- Our model: *vegetableCompartmentTemperature*

The property created by humans is shorter and understandable however to describe a property of a refrigerator, the word *compartment* is important and should not be eliminated.

A bad case is as followings:

- Input: *remaining stored electricity 2*
- Human *remainingCapacity2*
- Our model: *remainingStoredElectricity*

The machine-generated property name is semantically better than the human-created one. However, the number *2* is important to identify the property. To handle other cases, an effort to create rules for word choice is required.

A typical fair case is as followings:

- Input: *illuminance level*
- Human: *brightness*
- Our model: *illuminanceLevel*

This is a fair case because the generation of synonyms is not yet supported by the NLP model. However, it is a rare case because this is the only error from the Japanese-English translation of the EDO. The Human-Created result is the patch for this error since the EDO has been already published as an international standard.

Compared with TF-IDF, the propose model makes difference in long inputs such as:

- Input: *Measured instantaneous charging electric energy*
- TF-IDF: *measuredInstantaneousElectricEnergy*
- Our model: *instantaneousChargingElectricEnergy*

As TF-IDF focuses on rare words rather than frequent ones, it changes the overall meaning of this case. Otherwise, BERT considers semantic context of each word against the whole sentence's meaning, which makes the output understandable and concise.

V. DATA MODEL GENERATORS

To enhance the reusability and the extensibility of the *eModelGen*, the *Semantic Analyzer* and the *Model Generator* are decoupled. By implementing the corresponding generator, a target data model could be exported.

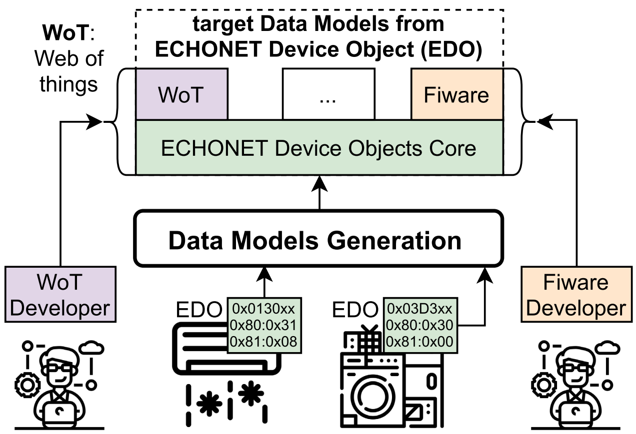

FIGURE 7. ECHONET device objects to platform specific data models.

In the scope of this paper, a data model for ECHONET Lite Web APIs (namely *Device Description*), a data model that supports ECHONET Lite in the FIWARE (namely *EL Smart Data Model*), and a data model that supports ECHONET Lite and the W3C Web of Things (namely *EL Thing Description*) integration are introduced. Both data models are based on JSON, however, XML, RDF [25], YANG [26], and ontology-based models could be extended in the same manner. The structure of JSON schemes of *Device Description* and *EL Smart Data Model* are summarized in Figure 8 and Figure 9 respectively.

Both of the JSON schemes share the same *deviceObjectName* and *propertyName* which are outputted from Section IV. The *data type* definition is the last piece of the puzzle.

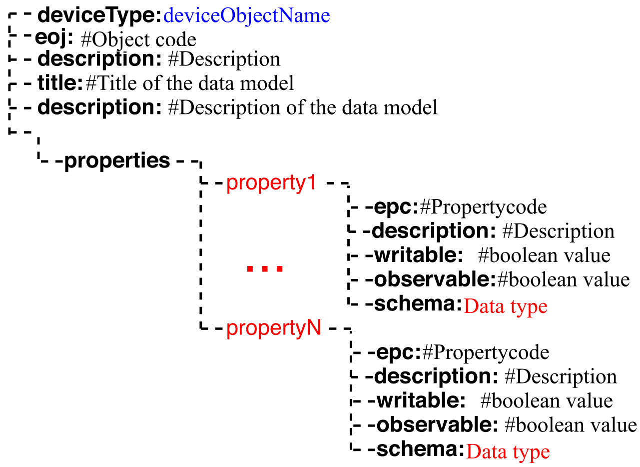

FIGURE 8. JSON schema of the data model for the ECHONET lite web APIs (*Device Description*).

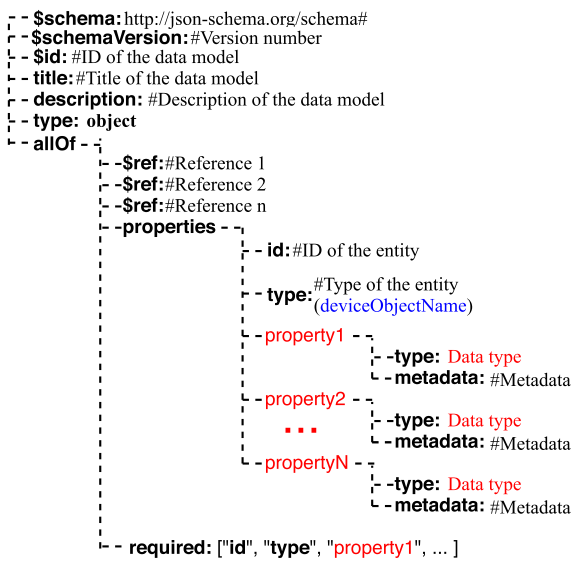

FIGURE 9. JSON schema of FIWARE smart data models (*EL Smart Data Model*).

The EDO includes data type definitions, however, those definitions are incompatible with the JSON-Schema [27] definitions which are being used to provided data formats for most of the current Web APIs. Since data type formats of the JSON-Schema are also supported in the FIWARE ecosystem and ECHONET Lite Web APIs, *data type* fields (in Figure 8 and Figure 9) are mapped from the ECHONET device objects specification as in Table 5.

Because the target data models are in the form of JSON schemes, JSON serializers are created using a Java library, namely *JSON.simple*. As a result, the total number of **117** JSON schemes of each data model has been generated within several seconds and **100%** of generated schemes have been passed the JSON syntactic checker. The *EL*

TABLE 5. Example of ECHONET device objects to data models mapping rules.

		Device Description	Fiware Data Model	Thing Description
Device Object	eoj	eoj	customTag:eoj	customTag:eoj
	className	descriptions	description	descriptions
	shortName	deviceType	title	title
Property Object	epc	epc	customTag:epc	customTag:epc
	ppName	descriptions	description	descriptions
	note	note	x	x
	atomic	atomic	customTag: atomic	customTag: atomic
	shortName	propertyName	propertyName	propertyName
	accessRule	Get/Set/Observable	Get/Set/Observable	Get/Set/Observable
Data Type Object	DateTime	type: String format: date-time	type: String format: date-time	type: String format: date-time
	Time	type: String format: time	type: String format: time	type: String format: time
	Date	type: String format: data	type: String format: date	type: String format: date
	Raw	type: String	type: Text	type: String
	Object Bitmap	type: Object	type:StructureValue	type:Object
	Array	type: Array	type: Array	type: Array
	Number Level Numeric	type: Number	type: Number	type:Number
	State	type: Boolean (on/off) type: String (otherwise)	type: Boolean (on/off) type: String (otherwise)	type: Boolean (on/off) type: String (otherwise)
	Null	type: null	x	type: null

Thing Description, which is available at the W3C Github,[3] was tested with 100% success rate at the 2021 September Plugfest/Testfest organized by the W3C [28].

VI. CONCLUDING REMARKS

To keep up with the spreading of Web API, data models to support the Web API are creating and publishing by the ECHONET Consortium. In order to contribute that effort, this paper introduced a solution to generate data models for the ECHONET Lite protocol from the ECHONET Device Object specification. Instead of hundreds of working hours of experts from the ECHONET Consortium, the proposed solution is able to generate target data models within several seconds using a normal personal computer.

The proposed solution implements (i) a *Semantic Analyzer* to learn and imitate the way that experts are using to create vocabularies from description sentences by utilizing NLP techniques and (ii) rule-based *Data Model Generators* to generate data models from a set of vocabularies in a fast and consistent manner. By decoupling those building blocks, the reusability and extensibility in supporting data models for various platforms are assured.

The proposed semantic analyzer is efficient to extract the most meaningful words to create property names and object names from the ECHONET device objects specification. By taking advantage of BERT and the unsupervised-learning approach, our model gains less effort than the traditional naming approaches while the quality of generated phrases is equivalent to human-generated outputs. Through experimental results, this work reveals the appropriate threshold $b = 80$ with the novel cost function for evaluating the information ratio. Experimental results show that definitions generated by the machine are better than humans in preserving the semantic of original descriptions while it is shorter. Nevertheless, there are cases where only semantic is not enough, and the verification, as well as corrections by experts, are needed at the last step to release data models. Therefore, the NLP model could be utilized as a recommendation system.

The rule-based *Data Model Generators* has beat humans in syntactic exporting because machines have the ability to work tirelessly speedily, and consistently. In this paper, two JSON serializers have been implemented to export data models for ECHONET Lite Web APIs and FIWARE Smart Data Models. As a result, within several seconds, **117 JSON schemes** (100% of the specification) for each data model have been exported and 100% of the generated JSON schemes pass the syntactic checker.

The extension of an ontology serializer to export the s4echonet ontology, a SAREF extended ontology for the ECHONET Device Object, is desired as future work.

[3]https://github.com/w3c

ACKNOWLEDGMENT

The authors would like to thank members of the ECHONET Web API working group for their effort in providing input data, comments, and feedbacks.

REFERENCES

[1] F. Curbera, M. Duftler, R. Khalaf, W. Nagy, N. Mukhi, and S. Weerawarana, "Unraveling the web services web: An introduction to SOAP, WSDL, and UDDI," *IEEE Internet Comput.*, vol. 6, no. 2, pp. 86–93, Mar. 2002.

[2] S. Vinoski, "RESTful web services development checklist," *IEEE Internet Comput.*, vol. 12, no. 6, pp. 95–96, Nov. 2008.

[3] W. Tan, Y. Fan, A. Ghoneim, M. A. Hossain, and S. Dustdar, "From the service-oriented architecture to the web API economy," *IEEE Internet Comput.*, vol. 20, no. 4, pp. 64–68, Jul. 2016.

[4] V. Charpenay, T. Kamiya, S. Kabisch, M. Kovatsch, and M. McCool. (Apr. 2020). *Web of Things (WoT) Thing Description*. W3C Recommendation. [Online]. Available: https://www.w3.org/TR/2020/REC-wot-thing-description-20200409/

[5] *Service Diagnostic Interface for Consumer Electronics Products and Networks—Implementation for Echonet*, Standard IEC62394:2022, International Electrotechnical Commission, Geneva, Switzerland, 2022. [Online]. Available: https://webstore.iec.ch/publication/66914

[6] (Mar. 2020). *Detailed Requirements for Echonet Device Objects*. [Online]. Available: https://echonet.jp

[7] E. Cambria and B. White, "Jumping NLP curves: A review of natural language processing research [review article]," *IEEE Comput. Intell. Mag.*, vol. 9, no. 2, pp. 48–57, May 2014.

[8] R. G. Castro, M. Lefrancois, M. P. Villalon, and L. Daniele, "The ETSI SAREF ontology for smart applications: A long path of development and evolution," in *Energy Smart Appliances: Applications, Methodologies, and Challenges*. Hoboken, NJ, USA: Wiley, 2023, pp. 183–215.

[9] *Requirements and Architecture of the Home Energy Management System and Home Network Services*, Telecommunication Standardization Sector of ITU, Geneva, Switzerland, Jan. 2015.

[10] V. C. Pham, Y. Lim, A. Sgorbissa, and Y. Tan, "An ontology-driven ECHONET lite adaptation layer for smart homes," *J. Inf. Process.*, vol. 27, pp. 360–368, Sep. 2019.

[11] S. Hanke, C. Mayer, O. Hoeftberger, H. Boos, R. Wichert, M.-R. Tazari, P. Wolf, and F. Furfari, " universAAL—An open and consolidated AAL platform," in *Ambient Assisted Living*, Berlin, Germany, 2011.

[12] H. Kodama, "The ECHONET lite specifications and the work of the ECHONET consortium," *New Breeze-Quart. ITU Assoc. Jpn.*, vol. 27, no. 2, pp. 4–7, Apr. 2015.

[13] M. Umejima, "Japan's power meter deployment with ECHONET lite over IPV6," *New Breeze-Quart. ITU Assoc. Jpn.*, vol. 27, no. 2, pp. 12–13, Apr. 2015.

[14] H. Fujita, H. Sugimura, M. Hamamoto, and M. Isshiki, "Improvement of the device descriptions of ECHONET lite by adding version specific information," in *Proc. IEEE 8th Global Conf. Consum. Electron. (GCCE)*, Oct. 2019, pp. 791–793.

[15] H. Ed-Douibi, J. L. C. Izquierdo, and J. Cabot, "Example-driven web API specification discovery," in *Modelling Foundations and Applications*, A. Anjorin and H. Espinoza, Eds. Cham, Switzerland: Springer, 2017, pp. 267–284.

[16] F. Cirillo, G. Solmaz, E. L. Berz, M. Bauer, B. Cheng, and E. Kovacs, "A standard-based open source IoT platform: FIWARE," *IEEE Internet Things Mag.*, vol. 2, no. 3, pp. 12–18, Sep. 2019.

[17] H. Li, D. Seed, B. Flynn, C. Mladin, and R. Di Girolamo, "Enabling semantics in an M2M/IoT service delivery platform," in *Proc. IEEE 10th Int. Conf. Semantic Comput. (ICSC)*, Feb. 2016, pp. 206–213.

[18] (Aug. 2020). *Echonet Lite Web API*. [Online]. Available: https://echonet.jp/

[19] M. Allamanis, E. T. Barr, C. Bird, and C. Sutton, "Learning natural coding conventions," in *Proc. 22nd ACM SIGSOFT Int. Symp. Found. Softw. Eng.*, New York, NY, USA, Nov. 2014, pp. 281–293, doi: 10.1145/2635868.2635883.

[20] R. Wang, H. Zhang, G. Lu, L. Lyu, and C. Lyu, "FRET: Functional reinforced transformer with BERT for code summarization," *IEEE Access*, vol. 8, pp. 135591–135604, 2020.

[21] E. W. Høst and B. M. Østvold, "Debugging method names," in *ECOOP 2009—Object-Oriented Programming*, S. Drossopoulou, Ed. Berlin, Germany: Springer, 2009, pp. 294–317.

[22] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1. Minneapolis, MN, USA, Jun. 2019, pp. 4171–4186.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. U. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2017, pp. 5998–6008.

[24] A. Rajaraman and J. D. Ullman, *Data Mining*. Cambridge, U.K.: Cambridge Univ. Press, 2011, pp. 1–17.

[25] S. Decker, S. Melnik, F. van Harmelen, D. Fensel, M. Klein, J. Broekstra, M. Erdmann, and I. Horrocks, "The semantic web: The roles of XML and RDF," *IEEE Internet Comput.*, vol. 4, no. 5, pp. 63–73, Feb. 2000.

[26] M. Bjorklund, "YANG—A data modeling language for the network configuration protocol (NETCONF)," Internet Eng. Task Force (IETF), USA, Tech. Rep. RFC6020, 2010.

[27] F. Pezoa, J. L. Reutter, F. Suarez, M. Ugarte, and D. Vrgoč, "Foundations of JSON schema," in *Proc. 25th Int. Conf. World Wide Web*, Geneva, Switzerland, Apr. 2016, pp. 263–273, doi: 10.1145/2872427.2883029.

[28] V. C. Pham, T. Xin, M. Sioutis, and Y. Tan, "Bridging the ECHONET lite protocol and the W3C web of things for smart homes in Japan," in *Proc. IEEE 11th Global Conf. Consum. Electron. (GCCE)*, Oct. 2022, pp. 781–785.

VAN CU PHAM (Member, IEEE) received the B.Eng. degree in software engineering from FPT University, in 2014, and the M.S. and Ph.D. degrees in information science from the Japan Advanced Institute of Science and Technology (JAIST), in 2016 and 2020, respectively. Currently, he is a Research Assistant Professor with the Center for Digitalization Endeavors, JAIST. His research interests include the cross-domain interoperability of smart systems, smart infrastructure, home network systems, home network management, and network simulation. He is a member of IPSJ.

NGUYEN THI DIEU LINH received the Ph.D. degree from the Harbin Institute of Technology, China, in 2013. She is currently the Deputy Head of the Department of Science and Technology, Hanoi University of Industry (HaUI), Vietnam. She has more than 20 years of academic experience in electronics, the IoT, smart homes, smart gardens, and telecommunications. She has authored or coauthored many research papers that are published in journals, books, and conference proceedings. She teaches graduate and master's level courses with HaUI.

TUNG LE received the B.S. degree (Hons.) in computer science from the University of Science, Vietnam National University, Ho Chi Minh City, Vietnam, in 2012, the M.S. degree in information science from the Japan Advanced Institute of Science and Technology (JAIST), in 2018, and the Ph.D. degree in information science from the School of Information Science, JAIST, in 2021. He is currently a Lecturer in information technology with the Ho Chi Minh City University of Science (HCMUS), Vietnam. His current research interests include information retrieval, deep learning, natural language processing, visual question answering, and multi-modal systems.

TIEN HUY NGUYEN received the B.S. degree in software technology and the M.S. degree in computer science from the Ho Chi Minh City University of Science (HCMUS), Vietnam, in 2010 and 2015, respectively, and the Ph.D. degree in information science from the School of Information Science, Japan Advanced Institute of Science and Technology (JAIST), in 2019. He is currently a Lecturer in computer science with HCMUS. He leads the Laboratory on Natural Language Processing and Speech Processing, Zalo, VNG. His current research interests include information retrieval, deep learning, speech recognition, and text-to-speech.

YASUO TAN (Member, IEEE) was born in 1965. He received the Ph.D. degree from the Tokyo Institute of Technology, in 1993. In 1993, he joined as an Assistant Professor with the School of Information Science, Japan Advanced Institute of Science and Technology (JAIST). He has been a Professor, since 2007. His research interests include the IoT systems, especially smart home systems. He is a member of the Information and Communication Council, ACM, IPSJ, IEICE, IEEJ, JSSST, and JNNS, an Advisory Fellow of the ECHONET Consortium, the Chair of the Technology Standards Subcommittee of the Smart IoT Acceleration Forum, and the Chairman of Green Grid Platform at Home Alliance.