이종 IoT 데이터 표현을 위한 그래프 모델: 스마트 캠퍼스 관리 사례 연구

뉘엔반퀴엣, 뉘엔휴쥐, 뉘엔양쯔엉, 김경백
전남대학교 전자컴퓨터공학부
e-mail : quyetict@utehy.edu.vn, huuduyce@gmail.com, truongnguyengiang.bk@gmail.com,
kyungbaekim@.jnu.ac.kr

A Graph Model of Heterogeneous IoT Data Representation : A Case Study from Smart Campus Management

Van-Quyet Nguyen, Huu-Duy Nguyen, Giang-Truong Nguyen, Kyungbaek Kim
Department of Electronics and Computer Engineering, Chonnam National University

Abstract

In an Internet of Thing (IoT) environment, entities with different attributes and capacities are going to be connected in a highly connected fashion. Specifically, not only the mechanical and electronic devices but also other entities such as people, locations and applications are connected to each other. Understanding and managing these connections play an important role for businesses, which identify opportunities for new IoT services. Traditional approach for storing and querying IoT data is used of a relational database management system (RDMS) such as MySQL or MSSQL. However, using RDMS is not flexible and sufficient for handling heterogeneous IoT data because these data have deeply complex relationships which require nested queries and complex joins on multiple tables. In this paper, we propose a graph model for constructing a graph database of heterogeneous IoT data. Graph databases are purposely-built to store highly connected data with nodes representing entities and edges representing the relationships between these entities. Our model fuses social graph, spatial graph, and things graph, and incorporates the relationships among them. We then present a case study which applies our model for representing data from a Smart Campus using Neo4J platform. Through the results of querying to answer real questions in Smart Campus management, we show the viability of our model.

1. Introduction

In an Internet of Thing (IoT) environment, a variety of different entities (e.g., devices, people) are connected to each other [1]. The major challenges for IoT applications are storing and analyzing data from a huge number of 'things' with deeply complex relationships. Using relational database management systems (RDMS) such as MySQL or MSSQL may not be flexible and sufficient for handling IoT data because nested queries and complex joins on multiple tables are often required for such highly connected data. Fortunately, graph databases are purposely-built to store highly connected data with nodes representing entities and edges representing relationships between these entities. Moreover, graph databases are supported by a lot of graph query languages tools such as Neo4J [2] or Pregel [3], with a variety of query types such as regular path query [4][5] or reachability path query [6]. Thus, building a graph model for constructing a graph database of IoT data is particularly evident for storing and analyzing complex IoT data.

In recent years, many IoT applications, such as a smart campus and a smart car, have been widespread and considered as the future of Internet [7]. This paper considers the data management of a smart campus as a case study for building a graph model of IoT data representation. The definition of a smart campus is not unique, and it depends on the various aspects such as technology and infrastructure [8]. Here, we consider the aspect of infrastructure for constructing a graph database of a smart campus. To identify the entities which can be used in the graph, we consider five components in the infrastructure of a smart campus including smart library, smart room, smart health center, smart lighting, and smart parking. The things corresponding to each component could differ from others. In here, the challenge is how to fuse these different components and corresponding things together in only one graph. Moreover, in the management of a smart campus, the relations between people and things and the relations between locations and things are necessary to be considered. These relationship data may help manage a smart building efficiently and effectively with various applications such as network resource allocation and warning notification under emergency situations. In these applications, how to build and manage a social graph and a spatial graph as the components of a smart campus database is also a key challenge.

In this paper, we propose a graph model for constructing a graph database from IoT data. Our model fuses a social graph, a spatial graph, and a things graph into one graph model, and incorporates the relationships among them. We then present a case study which applies our model for representing data from a smart campus using Neo4J platform. Through the results of querying to answer the real questions in a smart campus management, we show the viability of our model.

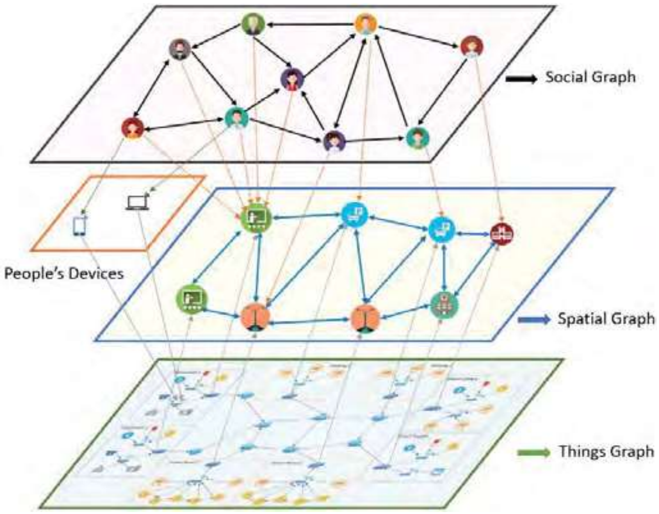


Figure 1. The proposed graph model of IoT data

2. Graph Modeling of IoT Data Representation

Our proposed graph model is depicted in Figure 1, which fuses social relations, location relations, things relations, and their interactions. The descriptions of nodes and edges in our graph model are described in Table 1 and Table 2, respectively. The graph components of the proposed model are explained as follows.

Things graph: This graph represents entities including sensors and devices and their connectivity. Each node represents a sensor or a device with different attributes such as SensorID, Name, Type, Position, Status, Timestamp, and Value. An edge represents the relationship between two sensors/devices, and two types of edge-label are used in things graph including *Connects* and *Links*.

Spatial Graph: This graph represents locations and their proximity. Each node is a place with the attributes such as *LocationID*, *PlaceName*, and *Coordinates*. Each edge indicates the proximity between two locations. Besides, a node in *Spatial Graph* could be connected by nodes in *Things Graph*, which indicates that some sensors/devices are employed at certain locations. This relation between a thing and a location is represented by using *AsignedTo* type edge. Also, a node in *Spatial Graph* could be connected by another node from *Social Graph* to show who is in a specific location. There are four edge types to represent these kinds of relations including *StayAt*, *StudiesAt*, *WorksAt*, and *TeachesAt*.

Social Graph: This graph represents people who join a smart campus. Each node is a person with some attributes such as *ID*, *Name*, *Age*, *Height*, *Weight*, and *Title*. An edge represents the relationship between two people, and it is assigned by using one of five edge types: *Friend*, *Colleague*, *Knows*, *Supervisor*, and *LeaderOf*. Furthermore, a node of *Social Graph* could be connected to a node from *Spatial Graph* to show where a person is and connected to a node from *Things Graph* to indicate which things are used by a person.

By using three types of nodes and fourteen types of edges, we make many things in a smart campus being connected on one big graph, and it provides an opportunity for storing and analyzing IoT data by using various graph based techniques.

Table 1. Node Types Description

Node Type	Description	Attributes
People	Teachers, Students, Staffs	ID, Name, Age, Height, Weight, Title, Residence, GSP Location
Locations	Classrooms, Library, Health Center, Parking Places	LocationID, Place Name, Type, Address, Coordinates
Things (Sensors)	Light Sensors, Temperature Sensors, Fire Sensors	SensorID, Name, Type, Position, Status, Timestamp, Value
Things (Devices)	Routers, Switches, Access Points, Labtops, Phones, Cameras, Other Devices	DeviceID, Name, Type, Position, Status

Table 2. Edge Types Description

Edge Type	Description	Relations
Friend, Colleague, Knows, Supervisor, LeaderOf	Student – **Friend** → Student Staff – **Colleague** → Staff Person – **Knows** → Person Teacher – **Supervisor** → Student Staff – **LeaderOf** → Staff	Person - Person
StudiesAt, TeachesAt, WorksAt StayAt	Student – **StudiesAt** → Classroom/Library Teacher – **TeachesAt** → Classroom Staff – **WorksAt** → Office Person – **StayAt** → Road/Health Center/Parking Place	Person – Location (Activities)
Owns	Person – **Owns** → Device	Person - Thing
Connects, Links	Sensor/Device – **Connects** → Access Point Switch – **Connects** → Router Router – **Links** → Router	Thing - Thing
AssignedTo	Thing – **AssignedTo** → Location	Thing - Location
Proximity	Location – **Proximity** → Location	Location - Location

3. A Case Study from Smart Campus Management

In this section, we present a case study that applies our graph model to represent heterogeneous IoT data from a smart campus. First, we describe how to collect IoT data for the proposed graph model. Then, we present the way to storing the collected data into a graph database. Finally, we present how to answer some real questions in a smart campus management by using a graph query language, Cypher.

3.1. Collecting IoT Data for a graph model

IoT data for a graph model can be collected by using both offline mode and real-time mode. With offline mode, IoT data is manually collected by the human. In contrast, with real-time mode, the data is automatically collected by using supporting tools or dedicated software/techniques.

For the data of a things graph, the values of the attributes of a sensors/device such as *SensorID/DeviceID*, *Name*, *Type*, and *Position* will be put into graph database only once by using offline-mode (e.g., put data from *.csv file into a graph database), except the information from mobile sensors /devices which suddenly appear or are moved out from the system. The ones such as *Status*, *Timestamp*, and *Value* are needed to be updated in a real-time manner. Also, the edges among stationary device nodes can be generated manually based on the facts of the connection; meanwhile, the edges which are generated/removed from mobile device nodes should be recognized in a real-time manner.

For the data of a spatial graph, the information of both nodes and edges can be easily assembled in offline mode by using white papers, public data from a campus, or some data (e.g., Place Name, Coordination) from third parties such as Google Place.

For the data of a social graph, each person who is a teacher/student/visitor on a campus becomes a node in the graph. Their personal information (node attributes) and their relationships can be fused from multiple sources. While public records about people, who are working or studying in the campus, provide the basic information (e.g., name, email, phone number, address, etc.) for constructing nodes and edges in the social graph; social media data from social network applications (e.g., Facebook, Twitter, LinkedIn, etc.) are often necessary to enrich the graph information. For example, by exploring social media activities in Facebook we can create an edge to represent who knows whom via adding a friend, or update node information with Location attribute to indicate where a person has checked-in. Here, matching/fusing information between nodes which are constructed from public records and the information from social media data is a key challenge to build a social graph for a smart campus. We consider this challenge as one of our future research.

3.2. Storing IoT Data into a graph database

In this case study, we use Neo4J as a graph database system for storing as well as querying our graph data. For that, we prepare graph data in CSV files format based on the proposed graph model. As a simple case, we consider to store a things graph, in which nodes represent light sensors, temperature sensors, smoke sensors, classrooms, access points, routers, and switches; edges represent the connection among them. The data in CSV files are loaded into Neo4J to make a unique graph as visualized in Figure 3. Note that, node attributes are hidden, except node ID, edges are labeled by a word likes *CONNECT* or *LINK*.

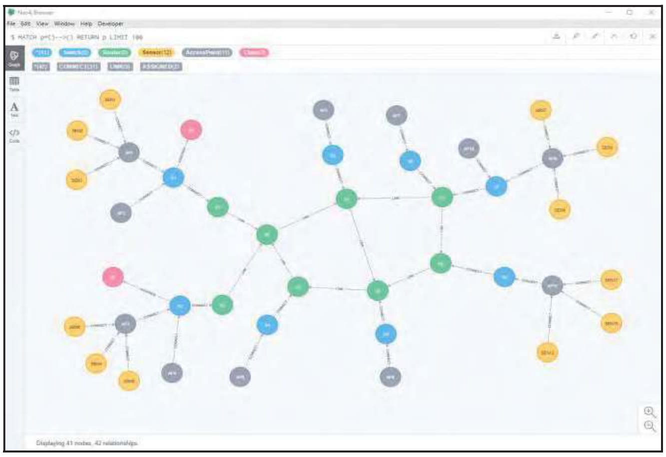


Figure 3. Visualization of IoT data by using graph database with Neo4J

3.3. Querying information from graph database

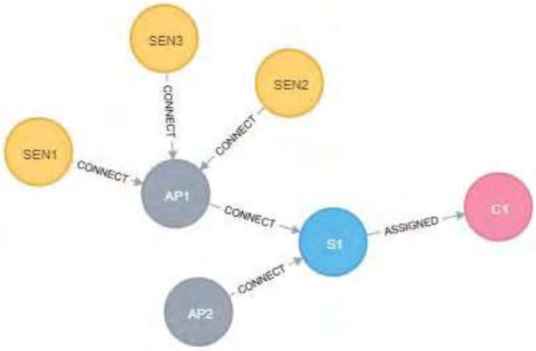
As an efficient way to understanding the status of many things in a smart campus, we use queries to access the proposed graph database. A common type of graph query is a regular path query, whose answer is a set of tuples of nodes connected by paths corresponding to a given regular expression [4]. This kind of query is supported by most graph query languages. In this case study, we use Cypher - a declarative query language developed by Neo4J Inc., which allows users to make graph queries like “SQL for graphs” to find out the data that they want to get. We present how to answer three real questions in a smart campus management by using graph queries with Cypher as shown in Table 3. While the first question can be answered by a simple query whose answer contains a set of nodes only, the other ones are more complex with path matching and recursive query whose answers are a path or a subgraph, as shown in Figure 4.

Table 3. Answering questions in Smart Campus management based on graph database using Neo4J platform

Query	Description
1	*Which sensors are not working?*
	MATCH (s:Sensor {status: 'Not Working'}) RETURN s;
2	*Which light sensors are having problem in classrooms where students are studying?*
	MATCH (sensor {type: 'LIGHT', value: 'null'})-[:CONNECT]->(ap)-[:CONNECT]->(switch)-[a:ASSIGNED]->(c:Class) WHERE c.num_student > '0' RETURN sensor,ap,switch,c;
3	*How are the “things” connected in the classroom ‘B6-105’?*
	MATCH (thing)-[:CONNECT*]->(switch)-[a:ASSIGNED]->(c:Class {room: 'B6-105'}) RETURN thing,switch,a,c

(a) Result of Query 1

(b) Result of Query 2



(c) Result of Query 3

Figure 4. The result of three graph queries

4. Conclusion

In this paper, we proposed a graph model for constructing a graph database from IoT data. Our model fuses a social graph, a spatial graph, and a things graph, and incorporates the relationships among them. We also presented a case study which applies our model for representing IoT data from a smart campus using Neo4J platform. Through the results of querying to answer the real questions in a smart campus management, we showed the viability of our model.

Acknowledgment

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT & Future Planning(NRF-2017R1A2B4012559). This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2018-2016-0-00314) supervised by the IITP(Institute for Information & communications Technology Promotion).

References

[1] Arora, Vaibhav, Faisal Nawab, Divyakant Agrawal, and Amr El Abbadi. "Multi-representation based data processing architecture for IoT applications." In Distributed Computing Systems (ICDCS), 2017 IEEE 37th International Conference on, pp. 2234-2239. IEEE, 2017.

[2] Malewicz, Grzegorz, et al. "Pregel: a system for large-scale graph processing." Proceedings of the 2010 ACM SIGMOD International Conference on Management of data. ACM, 2010.

[3] Neo4j Inc., "The Neo4j Graph Platform", 2018. [Online]. Availible: https://neo4j.com. [Accessed: 02- Sept- 2014].

[4] Nguyen, Van-Quyet, and Kyungbaek Kim. "Estimating the Evaluation Cost of Regular Path Queries on Large Graphs." In Proceedings of the Eighth International Symposium on Information and Communication Technology, pp. 92-99. ACM, 2017.

[5] Nguyen, Van-Quyet, and Kyungbaek Kim. "Efficient Regular Path Query Evaluation by Splitting with Unit-Subquery Cost Matrix." IEICE TRANSACTIONS on Information and Systems100.10 (2017): 2648-2652.

[6] Nguyen, Van-Quyet, Le-Duc Tung, and Zhenjiang Hu. "Minimizing data transfers for regular reachability queries on distributed graphs." Proceedings of the Fourth Symposium on Information and Communication Technology. ACM, 2013.

[7] Sari, Marti Widya, Prahenusa Wahyu Ciptadi, and R. Hafid Hardyanto. "Study of Smart Campus Development Using Internet of Things Technology." In IOP Conference Series: Materials Science and Engineering, vol. 190, no. 1, p. 012032. IOP Publishing, 2017.

[8] Muhamad, Wardani, Novianto Budi Kurniawan, and Setiadi Yazid. "Smart campus features, technologies, and applications: A systematic literature review." Information Technology Systems and Innovation (ICITSI), 2017 International Conference on. IEEE, 2017.