

How to Structure Data from an IoT Monitoring Network¹

Ziwen Yu²

The Internet of Things, or IoT, refers to the billions of physical devices (e.g., smartphones) that collect and share data through the Internet. This publication discusses the typical attributes in IoT monitoring networks and presents a data structure that outlines best practices in organizing data. The main goal is to provide knowledge on how such a system's data should be structured so that these IoT monitoring network owners, such as agricultural producers, can optimize how they use their data. Thus, the primary audience is agricultural producers, consultants, and others who install, manage, or use such IoT monitoring systems for decision making.

What is an IoT monitoring network?

A network of sensors measuring temperature, precipitation, wind speed, and similar weather-related information is a typical monitoring system used in agriculture. However, it would be very complicated and costly to achieve high monitoring coverage if the data transferring between sensors and the central server is built on wires. IoT is a lifesaver in this case that realizes the wireless data communications through the Internet and allows sensors to be deployed remotely at locations that were previously difficult to monitor. In other words, the IoT monitoring system is a network of IoT-based sensors that transfer the monitoring

data to the server or other terminals through the Internet. Such a system has been widely used in agriculture to increase yield, quality, and profit (Talavera et al. 2017). Using cameras for livestock monitoring, ranchers can gather data regarding the health, well-being, and location of their cattle. Smart irrigation systems can use IoT soil sensors to monitor the soil moisture and temperature as the foundation for the proper irrigation schedule. An IoT weather station can be equipped with many sensors to measure temperature, relative humidity, wind speed, and other climate characteristics to provide comprehensive information in helping plan agricultural operations.

Advantages and Trends of IoT Monitoring in Agriculture

An IoT monitoring system is routinely adopted in agriculture operations to improve the efficiency and effectiveness of these operations. Such systems generate data at a high frequency, usually sub-hourly, with less labor, which enhances the ability to track the status of a farming operation. The reliability and accuracy of such data are relatively higher than those of the manually collected information. Thus, the decisions derived from this higher-standard data would, in turn, be more reliable and accurate in directing agriculture operations.

1. This document is AE548, one of a series of the Department of Agricultural and Biological Engineering, UF/IFAS Extension. Original publication date November 2020. Visit the EDIS website at <https://edis.ifas.ufl.edu> for the currently supported version of this publication.
2. Ziwen Yu, assistant professor, Department of Agricultural and Biological Engineering; UF/IFAS Extension, Gainesville, FL 32611.

The use of trade names in this publication is solely for the purpose of providing specific information. UF/IFAS does not guarantee or warranty the products named, and references to them in this publication do not signify our approval to the exclusion of other products of suitable composition.

The Institute of Food and Agricultural Sciences (IFAS) is an Equal Opportunity Institution authorized to provide research, educational information and other services only to individuals and institutions that function with non-discrimination with respect to race, creed, color, religion, age, disability, sex, sexual orientation, marital status, national origin, political opinions or affiliations. For more information on obtaining other UF/IFAS Extension publications, contact your county's UF/IFAS Extension office. U.S. Department of Agriculture, UF/IFAS Extension Service, University of Florida, IFAS, Florida A & M University Cooperative Extension Program, and Boards of County Commissioners Cooperating. Nick T. Place, dean for UF/IFAS Extension.

Benefited by wireless networks (e.g., cellular network and WiFi), monitoring equipment (e.g., sensors, batteries) can be deployed in a location which is remotely connected to farm infrastructure. In contrast, traditional monitoring systems are sometimes restricted by power and data connections. IoT-based monitoring systems with sensors that measure various parameters of the weather, ecosystem, and environment can be spatially distributed to promptly acquire a more comprehensive overall status of a farm, a state, and even a country. IoT tremendously reduces the cost of data communications among the sensors, servers, and users and enables wide applications of the monitoring system that were previously less practical. For example, many startup companies are developing advanced, low-cost, IoT-based soil sensors that can be deployed in a plug-in mode in many locations in a farm to measure soil status matrix and send the information to a data management platform remotely (Figure 1).



Figure 1. Sensoterra soil moisture sensors (www.sensoterra.com) from the Netherlands.
Credits: Sensoterra

Challenges before Data Interpretation

An IoT monitoring system can collect large amounts of data for decision making. Thus, there is a challenge with such large data sets in how the data are recorded and processed to keep the information complete and meet the dynamic requirements for future development and usage. Agricultural producers, data owners, and stakeholders who do not have the professional training in processing such large time-series data sets may not be able to use them effectively or efficiently. There are concerns with large amounts of data, such as how much storage space would be required on a hard drive, how much system resources would be consumed for high-frequent data recording and associated queries, and how efficient it would be

when querying on a table in billions of rows. While these concerns are valid, cloud databases and advanced technologies for large-storage hard drives can solve the storage issue. New time-series database platforms (e.g., fluxDB) accommodate high-frequency input and output. Data center technologies enable parallel operations that speed up complex query processes. Nevertheless, a remaining issue with data from IoT systems is how to organize the data so they can be efficiently recorded and used.

When determining how to organize and store data, attributes of the data should be considered. The readings from monitoring sensors are less useful without specifying additional corresponding attributes, such as location, units, and measurement types (e.g., air temperature, wind speed, air pressure), for each measurement. It is difficult to include and organize all attributes and readings in a uniform structure that can be stored in a single table. Firstly, some additional attributes are relatively static compared to sensor readings, which may cause huge redundancy, or large space, when added to each recorded data point. (See Table 1 for an example.) Secondly, these additional attributes, under the development of an IoT system, could even be dynamic themselves, which would require restructuring the table frame by adding new or inactivating old attributes. Either one would need substantial time and investment from IoT system managers, who do not have the necessary professional training and are not using any IT consulting service.

Data presentation is not data management

The data collected by sensors are typically numeric information with multiple attributes, including collection time and location, and other data descriptors. In a report or presentation, a subset of the whole data is usually tabulated in an intuitive fashion (Table 1) to make information delivery simple. Table 1 provides an example data set from an IoT system with three attributes and two measurements.

Such a table is usually large and includes all attributes of the system, even with duplicated information, as well as descriptive headers for each column. However, this usually misleads the public to believe that data management is merely putting everything together and providing column headings with self-explanatory texts. The way someone might present data in a report, such as in Table 1, would not be the same organization of data needed in a large database structure. Table 1 presents data in a way that is understandable by the general public. However, organizing data this way in a database system will cause issues, such as violating

certain naming policies in database systems, losing certain attributes information, and creating data redundancies.

In Table 1, the headers of measurement columns (e.g., soil temperature and air temperature) include detailed characteristics (e.g., measurement depth and units). While this is appropriate for sharing data in a report, this method would not be a best practice when storing the data in a database. The name conventions (Sarkuni 2014) that are commonly used in professional practices for most of the off-the-shelf databases indicate that column headings should always be letters, numbers, and underscores. Other special characters, such as dots, spaces, or dashes, are usually used as important signs for syntax, command separation, and calculation, which require extra and typically complex coding to differentiate between headers and signs. For example, dashes, such as the negative sign of the “Soil Temperature at -10 cm in °C,” may conflict with the calculation operator minus (-), if not specified differently in a query. In certain database platforms, some of these characters are not allowed to name columns because they may result in loss of attribute information.

Location is an important attribute for any measurement. Location is associated with not only where the data are collected, but also with other characteristics, such as the time zone, elevation of the site, and maintenance/active status. These characteristics, such as longitude and latitude, are normally static and replicated for each time step in the structure in Table 1. Such redundancy takes extra storage. Thus, a better method for organizing data would provide more efficient use of a database and reduce unnecessary redundancy.

The table structure in Table 1 is also limiting in terms of adaptability. If such a structure is used to store data, changes made to an IoT monitoring system (such as adding new sensors) may require significant efforts to alter the data structure.

Development Requirements

IoT monitoring systems are usually a system that develops over time such that the types of measurements collected and the space over which they are collected increase as the user identifies more needs and gains experience with the system. Such a feature of expandable size of system measurement and coverage is usually referred to as “scalable” (Shona and Arathi 2016). How to easily accommodate new locations and additional measurements that are added to a system is a critical question for the data management of an IoT system. In other words, the structure of a database

should be designed to minimize changes needed to add or remove new locations and measurements (Abu-Elkheir et al. 2013). See the following section for an example.

Best Practices for Data Management Structure

A desired data structure can be established by using relationships between different tables to reduce the data redundancies and keep the information complete while simplifying the operations for editing data attributes. Figure 2 and Tables 2, 3, and 4 illustrate a sample design and data of a weather monitoring system.

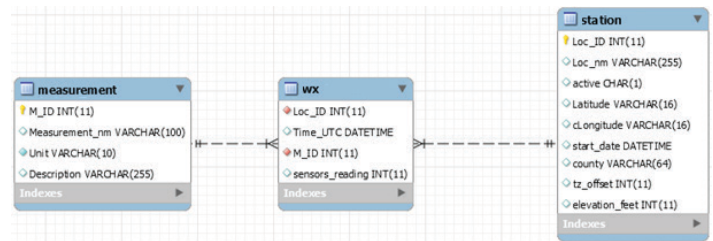


Figure 2. Diagram of database schema. A database schema is the skeleton structure that represents the logical view of the entire database. It defines how the data are organized and how the relations among them are associated.

Credits: This diagram is generated by MySQL, an open-source database platform, which is used for managing the data operations of the IoT monitoring system.

All information is categorized into three tables: station, measurement, and system_reading. The station table (Table 2) contains information about the locations where sensors are deployed. Its attributes include location name (Loc_nm), active status, latitude, longitude, start date, county, time zone offset, and elevation. An identity number, Loc_ID, is assigned to represent each location in the whole system, which also serves as the key to build a relationship of location between the station table and other tables (e.g., system_reading). Similarly, in the measurement table (Table 3), attributes of a measurement are listed. These include measurement name, unit, and description of measurement conditions. An identity, M_ID, is assigned for each measurement and used in other tables as a reference back to Table 3. Such relationships can be seen from the system_reading table (Table 4), storing actual sensor readings. These identities in Tables 2 and 3 represent the contents in the rest of the columns throughout the database. In other words, the full descriptors of a location, for example, only appear in one table and use its identities in the other tables. A column is defined in these tables to store the identities of interest and establish a relationship to the targeted table. Figure 2 illustrates these relationships by the dashed lines between tables. In this case, the system_reading table is

the hub that is related to the other two tables using Loc_ID and M_ID.

To explain how the data would be recorded in a system described by Figure 2 and Tables 2 to 4, here is an example of data collection and storage. A numerical sensor reading is collected with its time, location, and other data descriptors (e.g., units, the type of measurement, and the way the measurement was taken). In the system_reading table, the moment that a sensor reading is received is recorded with its Universal Time Coordinated (UTC). By tracing the time zone offset from the station table, the UTC time can be converted into local time corresponding to the Loc_ID to which the sensor reading belongs. For example, the local time for the last record in the system_reading table is actually “2/20/2020 00:00” after offsetting -5 from UTC time for Loc_ID 230. Thus, with the relationship built on Loc_ID, all other information in Table 2 can be traced. In the same manner, the descriptors of different measurements can be looked up by the M_ID from Table 3. For example, the sensor reading of 17 in the last row of Table 4 represents the “Soil temperature measured at 2 meters above ground” in “Celsius Degree,” according to the information in Table 3 referred to by M_ID of 1.

With the relationships across tables, the redundancy of station information is significantly reduced. Additionally, complicated measurement descriptors are combined in M_ID, which avoids potential information loss. However, the greatest improvement brought by this change is the separation of station and measurement with sensor readings. In this way, the IoT system development activities of adding and removing station or measurement can be simply performed by appending or deleting rows from these two tables without editing their columns or table structures. Thus, the data management efforts can be minimized.

Summary

While intuitiveness is key to delivering a message when presenting data, the associated data redundancy and complex descriptive column headers are not ideal for use in a database management system. The structure of the database of an IoT system must be static and meet the needs of the dynamic system development with potential editing of attributes, such as location and data descriptors, associated with sensor readings. The relationships among tables recording categorized data provide a good data organization solution that meets these requirements. Agricultural producers, consultants, and others who install, manage, or use such IoT monitoring systems for decision making can adopt these best practices in recording data

from IoT systems that will allow for more efficient use of storage space and data organization. More information on database management can be found in Dobson et al. 2018.

References

- Abu-Elkheir, M., M. Hayajneh, and N. A. Ali. 2013. “Data Management for the Internet of Things: Design Primitives and Solution.” *Sensors* 13(11): 15582–15612.
- Dobson, S., M. Golfarelli, S. Graziani, and S. Rizzi. 2018. “A Reference Architecture and Model for Sensor Data Warehousing.” *IEEE Sensors Journal* 18(18): 7659–7670.
- Sarkuni, S. 2014. “How I Write SQL, Part 1: Naming Conventions.” Accessed on September 28, 2020. <https://launchbylunch.com/posts/2014/Feb/16/sql-naming-conventions/>
- Shona, M., and B. Arathi. 2016. “A Survey on the Data Management in IoT.” *International Journal of Scientific and Technical Advancements* 2(1): 261–264.
- Talavera, J. M., L. E. Tobón, J. A. Gómez, M. A. Culman, J. M. Aranda, D. T. Parra, L. A. Quiroz, A. Hoyos, and L. E. Garreta. 2017. “Review of IoT Applications in Agro-industrial and Environmental Fields.” *Computers and Electronics in Agriculture* 142:283–297.

Table 1. Sample data from an IoT system that includes two measurements (soil temperature and air temperature) and three attributes (longitude, latitude, time).

Longitude	Latitude	Time	Soil Temperature at -10 cm in °C	Air Temperature at 10 cm in °C
-84.597	30.545	2/28/2020 15:00	13.35	15.03
-84.597	30.545	2/28/2020 14:45	13.27	14.87
-84.597	30.545	2/28/2020 14:30	13.11	14.54
-84.597	30.545	2/28/2020 14:15	12.9	14.37
-85.165	30.85	2/28/2020 14:00	17.76	14.59
-84.597	30.545	2/28/2020 14:00	12.66	14.21
-85.165	30.85	2/28/2020 13:45	17.64	14.34
-84.597	30.545	2/28/2020 13:45	12.38	13.95
-85.165	30.85	2/28/2020 13:30	17.4	14.1
-84.597	30.545	2/28/2020 13:30	12.06	13.61

Table 2. Sample data in station table (The bold row of data indicates an example explained in the document).

Loc_ID	Location	Active	Latitude	Longitude	Start_Date	county	tz_offset	elevation_ft
170	LIVE OAK	Y	30.303	-82.9	9/24/2002 0:00	SUWANNEE	-5	165
180	MACCLENNY	Y	30.282	-82.138	9/17/2002 14:00	BAKER	-5	126
230	BRONSON	Y	29.402	-82.587	9/24/2002 0:00	LEVY	-5	116

Table 3. Sample data in measurement table (The bold row of data indicates an example explained in the document).

M_ID	Measurement_nm	Unit	Description
1	temp_soil	C	Soil temperature measured at 10 cm below ground
2	temp_air	C	Air temperature measured at 60 cm above ground
3	temp_air	C	Air temperature measured at 2 m above ground

Table 4. Sample data in system_reading table (The bold row of data indicates an example explained in the document).

Loc_ID	Time_UTC	M_ID	Sensor_reading
170	2/20/2020 0:00	1	21
170	2/20/2020 0:00	2	18
180	2/20/2020 0:00	1	22
180	2/20/2020 0:00	2	18
230	2/20/2020 0:00	1	21
230	2/20/2020 0:00	2	18
230	2/20/2020 0:00	3	17