Master Course Photogrammetry and Geoinformatics, HFT Stuttgart

Visualization and Analysis of E-bike Usage in 3D City Model by Integration of Heterogeneous Sensor Data

By Thunyathep Santhanavanich

Abstract

According to the advances in information and communications technology, nowadays, the use of Internet of Things (IoT) has become a normal part of daily life as it allows interconnections among a wide variety of devices and sensors such as smartphones, smartwatches or any smart wearable devices, automobiles, or any object with a built-in sensor. However, these devices and sensors are developed by numerous different manufacturers which leads to the heterogeneity in the data management system. Consequently, a standard sensor protocol to sustain the interoperability among the various sensor systems is needed. In this research, the main contribution is building a sensor data management system for monitoring E-bike usages in the HFT i_city E-bike sharing project in the downtown area of Stuttgart city. The heterogeneous sensors in the project are including the SMART E-bike, the smartwatch Garmin Fenix X5, and the temperature data from OpenWeatherMap at the time this research is being conducted while the project is still open for more sensor devices and data sources. As for a scheme to tackle this challenge, three candidates of sensor protocols including 1) OGC SensorThings API, 2) OGC Sensor Observation Service, and 3) Advanced Sensor Data Delivery Service are implemented to manage the sensor from different sources in various format. (figure 1) Next, the performances by each sensor protocol are evaluated to find the best-qualified protocol. Last, all sensor data stored remotely by the selected protocol is used to develop a 3D web-based application for a visualization and analysis of the E-bike usage in 3D city model. (figure 2) To summarize the finding of this research, the OGC SensorThings API is the most competent IoT framework among the three candidate protocols. It serves all aspects including the IoT interoperability, understandability, and maintainability for developers and researchers which lead to the future expansion and development in terms of IoT devices and applications. It is also a successful sensor framework in the implementation for visualization and analysis of the E-bike usage in 3D City model by integration of heterogeneous sensor data.

Introduction

In this research, it aims to study and compare the existing sensor frameworks or standards on their capabilities to integrate the heterogeneous sensor systems together and provide the data in an efficient way. The challenge of this work is to maintain the interoperability from the different development layers, for example, the communication protocols and data models. Also, to solve the issues of the irregular time-series data streams from numerous IoT devices, the sensor network must provide the capability to aggregate or interpolate the data over time. For the data collection, the E-bike usage data from twelve volunteers in the area of the Stuttgart city [1] are used. Three sets of sensor data used in the study are including the following data: 1) E-bike sensors, 2) the Garmin Smart Watch, and 3) the open-source temperature data in the city of Stuttgart from “OpenWeatherData”. Then, to verify of the sensor network interoperability on both server and client side, the data from the studied sensor network are used to implement the 3D web-based application for the visualization and analysis of the E-bike usage in the 3D city model.

State of the Art

As the sensor technology, computer technology, and network technology are advancing together, the standards to link many diversities of technologies in an efficient way are needed. To address this, the Open Geospatial Consortium (OGC) has provided the Sensor Web Enablement (SWE) standards. With this standard, it allows users or developers to integrate many heterogeneous sensor systems with the accessible and manageable interface through the web. Many studies in topics of IoT networks use this standard as their sensor framework to manage diverse sensor data sources. Based on the previous works in several sensor protocol standards, a large number of researchers successfully used the OGC Sensor Observation Service (SOS) as their sensor management system for accessing sensor data and metadata since the first version of SOS has been approved in 2007. Still, some research finds this sensor framework is difficult in application development stage. [2] In contrast to the OGC SensorThings API, it is the most recent OGC standard for managing the sensor data and still has a low number of studies based on this standard [3], and this standard is developed based on the same OGC SWE with the improvement concept that overcomes the OGC SOS in many ways [4], for example, it supports the pagination and MQTT, a powerful machine-to-machine (M2M) connectivity protocol. Also, the main encoding of OGC SensorThings API in JSON with REST bindings which are more friendly and simpler for developers to understand, develop and expand the IoT application and keep its extensibility and maintainability. [5] However, the number of researches on sensor integration using OGC SOS are still higher than OGC SensorThings API. Moreover, both standards still have some gap in the aggregation function support which is the main reason that the sensor protocol “Advanced Sensor Data Delivery Service” is developed in a student project in HFT Stuttgart in 2016. [6] Accordingly, it can be stated that each protocol has its own advantages and disadvantages and all of them have been proved in the realistic projects on their interoperability and capability of heterogeneous sensors management. This thesis research will study and compare all three mentioned protocols including 1) OGC SensorThings API, 2) OGC Sensor Observation Service (SOS), and 3) Advanced Sensor Data Delivery Service (ASDDS) to find the best-qualified standard for managing heterogeneous E-bike usage data.

Methodology

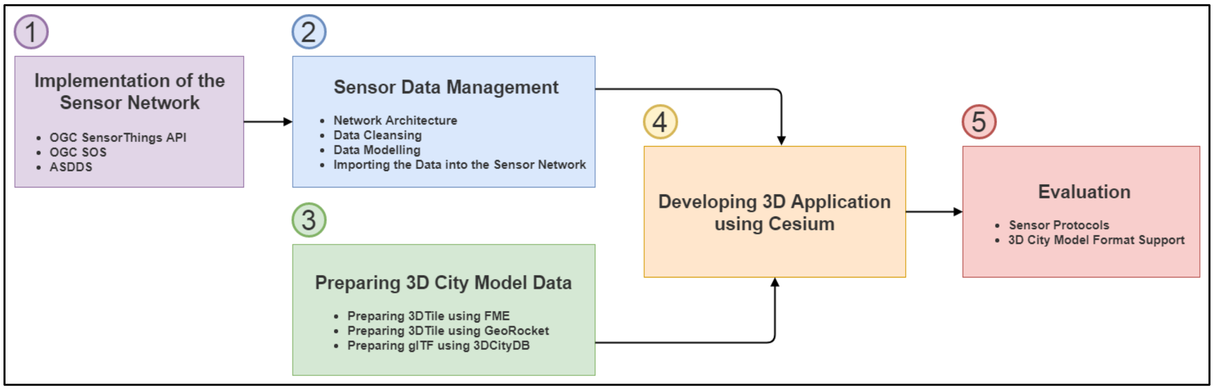

Figure 1: The overall research processes

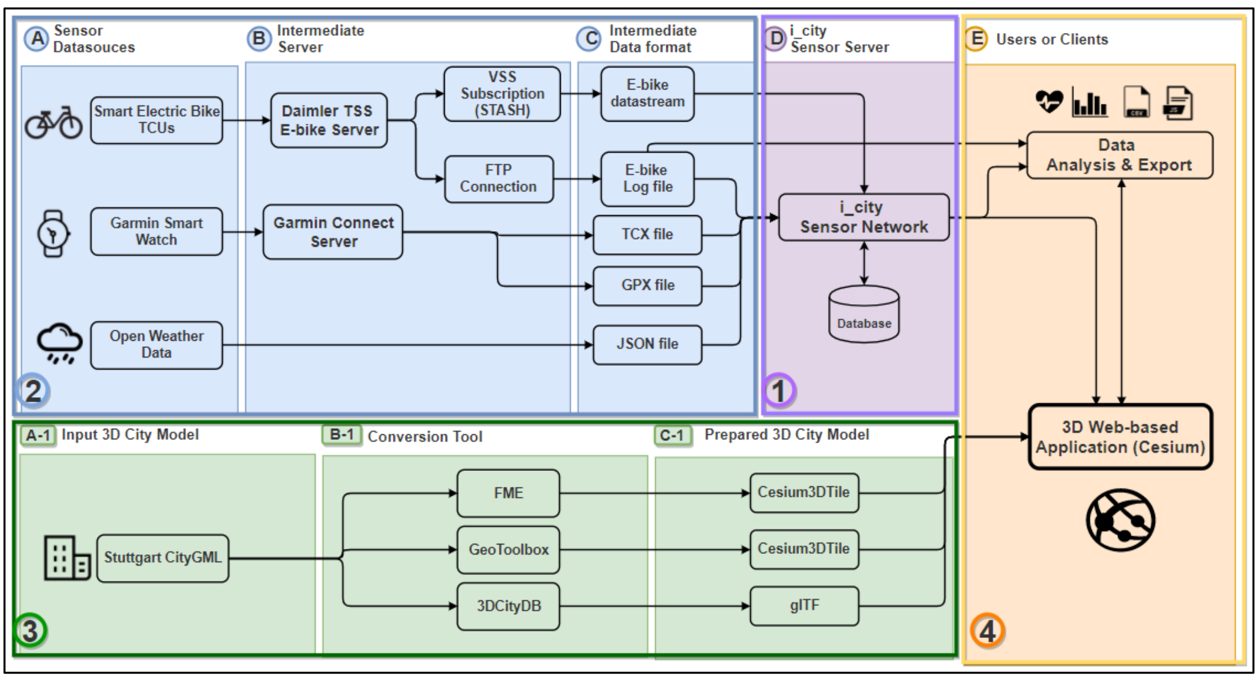

Figure 2: System Architecture

In this research, overall processes can be summarized as in Figure 1 which are divided into five main steps which are related to the final architecture of the project shown in Figure 2. In the first step, "① Implementation of the Sensor Network", the three selected sensor protocols are studied and implemented in the local server which is including 1) OGC SensorThings API, 2) OGC SOS, and 3) Advanced Sensor Data Delivery Service (ASDDS). The main goal of this step is to study and compare on their usages, limitations, capabilities or features in all sensor protocols. Then, the next step "② Sensor Data Management" is to manage the heterogeneous sensor data from E-bike TCU, Garmin Smart Watch, and other sensor data sources. In this step, the data management is starting with network architecture to link the data from different sources into the sensor network built form the first step ①. Then, the data cleaning is needed as each sensor data sources provide the redundant fields or entities in a different way. After that, the data modeling is done to match with the sensor protocol specification according to the first step ①. Finally, the prepared data will be imported into the sensor network. On the third step, "③ Preparing 3D City Model Data" is to prepare the 3D city model data in the downtown area of Stuttgart city which is provided in the CityGML format. As the CityGML is not able to be visualized on the Cesium application in the 3D environment directly. Thus, the conversions of the CityGML into Cesium 3D Tile or glTF are needed. The possible tools to do this conversion are including 1) FME 2017, 2) Georocket GeoToolbox 1.0.2, and 3) 3DCityDB. On the fourth step, "④ Developing 3D Application using Cesium", the main idea is to build a real use case application utilizing the heterogeneous sensor data from the prepared sensor network on step ① and ② together with the prepared 3D city model data from step ③. This application is called "i_city E-Bike sharing" (figure 3), it is a 3D web-based visual analytics application designed for researchers who are interested in the relation among the different parameters from sensor systems equipped on the E-bike sharing system in the Stuttgart city. It shows a simulation of the historical E-bike route track (figure 3) together with a comparison chart among the different selected parameters such as battery level, geospatial altitude, pedal force, motor support level, etc. (figure 4)

Figure 3: The "i_city ebike sharing" application screenshot showing the simulation of a historical E-bike route by integration the following sensor data from user: SMART E-bike, Garmin Smart Watch, and temperature data from OpenWeatherMap.

The application video demo is available online at ***https://www.youtube.com/watch?v=gEVmtgKRkPY***

Then, on the last step, "⑤ Evaluation" is to evaluate this research on the different implemented sensor protocols and the different methods to preparing the CityGML to visualize as a 3D city model on Cesium application.

Evaluation

In this research, three different sensor network protocols have been implemented and evaluated to select the most qualified protocol for monitoring the E-bike usage in a 3D application. The sensor data using in this research are including 1) the data from the SMART E-bike, 2) the smartwatch Garmin Fenix X5, and 3) the temperature data from OpenWeatherMap. The number of sensors - datastreams imported to the sensor network are shown in Table 1.

Table 1: Number of Datastreams using in this research

Date	Total Datastreams
22 November 2017	97,981
24 November 2017	167,193
27 November 2017	45,818
4 December 2017	89,313

To evaluate the technical performance on each sensor network protocol, the following metrics are used including 1) Request size of an operation, 2) Response time of an operation, 3) Response length of an operation, and 4) Support of dynamic 3D location of a moving sensor. Each metric had been tested on all the sensor protocols with the same dataset and the overall results of the evaluation are shown in table 2. The more details on the evaluation can be found in a full master thesis paper.

Table 1: Technical performance evaluation results

Evaluation Metrics	OGC Sensor Observation Service	OGC SensorThings API	Advanced Sensor Data Delivery Service
Request size of an operation	✓	✓✓	✓✓
Response Time of an Operation	✓	✓✓	✓✓
Response Length of an Operation	✓✓	✓	✓
Support of the Dynamic 3D Location of a Moving Sensor	✓	✓✓	**x**

(✓✓ = Good support, ✓ = support, x = not support directly and more development is needed)

Discussion

During the last six months, the three sensor network protocols: 1) OGC SensorThings API, 2) OGC Sensor Observation Service (SOS), and 3) Advanced Sensor Data Delivery Service (ASDDS) have been implemented and compared to find the most qualified protocol for managing the heterogeneous sensor systems in the i_city E-bike Sharing Project. On the implementation difficulty, the OGC SensorThings API has many sources of descriptions, practical application examples, and tutorials for new IoT developers in clear and understandable format. So that, it is the simplest framework to implement among all three candidate protocols. On the other hand, the OGC Sensor Observation Service has one critical disadvantage. Although it is based on the OGC's Sensor Web Enablement, which has been continually developing by a big group of researching for more than decades, it is the most difficult standards to learn. The complexity of the structure of the request body requirement in XML-base hinders the implementation of the project. On the other hand, the ASDDS structure is less complicated and can be implemented less difficulty. The fact that it is developed by HFT Stuttgart makes it easier to get some guidelines by personal contact.

For the database structure, both OGC SensorThings API and OGC Sensor Observation Service protocols are developed based on the OGC and ISO Observations & Measurements 2.0 standard. Their relational connections in the database structure can model and refer to the monitoring systems in the real world with a complete metadata. For, ASDDS, in contrast, some entities are missing, for example, the support entities to collect the geospatial location and the metadata of both sensors and observations. Moreover, the one-to-one entities relationship in the ASDDS impacts some of the repetitions of data on the server-side.

To apply the sensor protocol in the i_city E-bike sharing project, there was an unexpected issue of the E-bike's TCU sensor limitation that it will send the data through the server with the limited message length so that when the device sends a message to the server, some data attributes will not be sent in the same message but keep approximately 1-1.5 second delay and be sent as a separate message in order to keep the least message length communication. Accordingly, data will be sent to the server only the latest change value of the TCU sensor. For my finding, the OGC SensorThings API is the best IoT sensor protocol candidate to deal with this non-pattern irregular time-series data as the geospatial data in SensorThings API standard are collected in the "Locations" and "HistoricalLocation" entities which are independent of other observation results in "Observation" entities, so that the result of the updated geospatial location can be updated independently. On the other hands, the OGC SOS is collecting the data from moving sensor with the "OM_SpatialObservation" according to the "Spatial Filtering Profile" of the O&M standard. It always collects the observation result with the update of the location so that the locations will be sent only when there is an update of some other observation result. In conclusion, the SensorThings API can deal with this issue in a smooth way, while the ASDDS has no support of the geospatial locations but they must be added to the server as a new observation type in the network.

Throughout the application development phase, the OGC SensorThings API has proven to be a clear and efficient solution to receive the sensor data in the very compact request form. With a single HTTP request, users can flexibly request the very detailed response body of each observation with a query capability. For example, "$expand", "select", "$stop", "$filter", etc. For "$expand", it is used for requesting the result time, phenomenon time, latest and historical geospatial location, and all its meta data about things, sensors, and features of interest. For "$select", it is used for limiting the response body so that only some part will be shown. If users want to narrow the response by limiting only the highest or the lowest values over a specific period of time, they can use "$top" and "$filter". On the other hand, the OGC Sensor Observation Service always needs the large size of the request body to the server to request any message. For the ASDDS, the response is simple and similar to the OGC SensorThings API, however, it still lacks the capabilities to request for complex queries.

Considering the aggregation function over the sensor network, only the ASDDS protocol has this feature as its built-in function, but in this ASDDS version, it still supports only the "average" function with the limit option for the time-range selection. For the OGC SensorThings API and the OGC SOS, while they originally do not have this function, there are several methods to add it. The first way is to add the aggregation function output as a pre-computed value on the server-side. The second way is to add some programming library to compute this value on the client-side. The last way is to add an aggregation function on the server-side.

As the extensibility of the sensors is being examined, one disadvantage of the OGC SOS is found. This protocol needs a specific interface with a large request body for each sensor to add a new sensor or an IoT device to the network. In contrast, the OGC SensorThings API has proven the simple way to add a new sensor to the network with any simple HTTP request. Moreover, the OGC SensorThings can define physical any objects and any IoT sensors separately by specifying a "Thing" and a "Sensor". In other words, one physical object "Thing" can have one or many "Sensor(s)" and this protocol is able to detect each of them. For example, in future development, one E-bike can have not only TCUs, but also GNSS sensor, IMU, and any other IoT devices to observe many properties of the same E-bike at the same time.

Conclusion

This thesis research proposes the evaluation of the implementation of three different sensor protocols including 1) OGC SensorThings API, 2) OGC SOS, and 3) Advanced Sensor Data Delivery Service (ASDDS) to integrate the heterogeneous sensor systems together and the evaluation of data sufficiency provided by each protocol. To summarize the finding of this research, the OGC SensorThings API is the most competent IoT framework among the three candidate protocols. During the implementation process, the OGC SensorThings API has great quantities of documentation and tutorials. Also, the communication of OGC SensorThings API among IoT devices, servers, and clients is in a standards-based REST style and JSON-based encoding providing its advantage for developers to understand, implement, and develop an application based on this standard in a compact way. Additionally, it is the only protocol among the three candidate protocols that supports the MQTT communication which is very useful for adding a light-weight IoT device in the future. Even though the lack of the aggregation function is its downside, there are several methods to add this feature to this protocol including 1) adding an aggregation function on a client-side, 2) adding an aggregation function a server-side, and 3) pre-computation of the aggregated results on the server-side. In conclusion, the OGC SensorThings API sensor framework serves all aspects including the IoT interoperability, understandability, and maintainability for developers and researchers which lead to the future expansion and development in terms of IoT devices and applications. It is also a successful sensor framework in the implementation for visualization and analysis of the E-bike usage in 3D City model by integration of heterogeneous sensor data.

References

[1] L. Kohn, ““Erfassung emotionalen Erlebens von E-Bike Fahrenden mittels physiologischer Marker: eine Validierungsstudie im Forschungsprojekt „i_city“ der Hochschule für Technik Stuttgart in Kooperation mit Daimler TSS,”” Bachelor Thesis, Wirtschaftspsychologie, Hochschule für Technik Stuttgart, 2017.

[2] C. A. Henson, J. K. Pschorr, A. P. Sheth, and K. Thirunarayan, SemSOS: Semantic Sensor Observation Service. Piscataway, NJ: IEEE, 2009.

[3] S. Liang, T. Khalafbeigi, and C. Y. Huang, OGC SensorThings API: The official web site of the OGC SensorThings API standard specification. [Online] Available: https://github.com/opengeospatial/sensorthings. Accessed on: Nov. 02 2017.

[4] SensorUp Inc., Comparison Between OGC Sensor Observation Service And SensorThings API | SensorUp OGC SensorThings API Developer Centre. [Online] Available: http://developers.sensorup.com/webinars/Comparison-Between-OGC-Sensor-Observation-Service-And-SensorThings-API/. Accessed on: Oct. 01 2017.

[5] SensorUp Inc., Comparison between OGC Sensor Observation Service and SensorThings API. [Online] Available: https://www.slideshare.net/steve.liang/comparison-between-ogc-sensor-observation-service-and-sensorthings-api. Accessed on: Dec. 02 2017.

[6] M. Storz and H. Dastageeri, "Usage of Advanced Sensor Data Delivery Service (ASDDS)", personal communication, Oct. 2017.