The Requirements and Challenges of Visualizing Building Data

M. Nurminen*, A. Lindstedt*, M. Saari*, and P. Rantanen*
* Tampere University/Faculty of Information Technology and Communication Sciences, Pori, Finland
mikko.nurminen@tuni.fi

***Abstract* - Nowadays, energy consumption and especially energy saving are important issues. The news of global warming have increased the need to save energy in many areas of our living community. Visualization is an important trend in energy consumption research. Typical commercial buildings include building automation systems with special user interfaces are for automation professionals. There is a lack of UIs for information sharing with ordinary users. It has been proven that increasing consumer awareness can result in reduced energy consumption. Building automation systems, as well as other sensors installed in buildings collect a huge amount and variety of data related to heating, ventilation, and air conditioning. This paper presents a user interface for providing relevant data for the users, managers, and owners of buildings. Furthermore, requirements for visualizing building data are described. Finally, challenges faced in developing the visualization methods are discussed.**

***Keywords* - Visualization, user interface, building data, IoT**

I. Introduction

Considerable amount of energy is used to maintain comfortable living conditions in residential and commercial buildings. The energy is used for example in heating or cooling, lighting, and air conditioning of these buildings. In year 2019 in Finland 280.217 terajoules of energy was spent on the heating of buildings, accounting for 26% of all energy consumed that year.[1] With people's increased awareness of the need to lower energy usage to slow climate change, there rises a question: how can we aid the users, managers, and owners of buildings who are seeking to monitor the living conditions and reduce energy consumption? Earlier study [2] has shown that giving users real-time information about their energy usage led to less energy being consumed. We in the KIEMI research project embarked on a journey to create a system, DataSites, which would gather data from building data systems and would be able to visualize the gathered data to the users in an understandable way.

This paper presents the preliminary results of the KIEMI ("Vähemmällä Enemmän – Kohti Kiinteistöjen Energiaminimiä", or "Less is More: Towards Energy Minimum of Properties" in English) project, which aims to develop proof-of-concept demonstrations and prototype applications that illustrate how cost-effective, open, and modular solutions could be utilized to improve the energy efficiency of existing, older buildings.

Usually there are no User Interfaces accessible for normal users or residents of buildings, where they could view data from building systems. If building data UIs exist, they are usually supplier-specific, and only display data from that supplier's system. Also, they are reserved for managers of buildings only. To remedy this, DataSites' end-user client applications can be implemented as a web or mobile applications. A web browser or a smart phone are readily available and familiar to most people, and by using them as a platforms DataSites enables more people to discover relevant building data, and to make informed decisions on their own energy usage.

The rest of this paper is structured as follows. In Section II, the related studies on visualization are discussed. In Section III, the architecture of DataSites is examined. Then in Section IV the requirements placed on visualization of building data are specified. In Section V the Sites service and its UI are shown. Section VI closes this article with discussion and conclusions.

II. Related research

Feedback and visualization related to building data and energy consumption has attracted worldwide research interest. One important point of view is the awareness aspect of energy consumption, where the energy users' level of awareness and its effects on their energy consumption is studied. We made a survey of applications for apartment energy consumption monitoring [3] and one result was that several studies[4-7] show that knowledge of energy consumption increases efforts toward energy saving, leading to both monetary savings for the energy users, and less emissions from energy production, but this depends on the method used to produce the energy.

In an article [4] college students who received the real-time data of their energy consumption were more motivated and successful in reducing energy consumption than those students who received data after longer intervals. It is worth noting that both groups reduced their electricity usage when their awareness of it was increased. Another study [5] verified that an increase in received information about energy-saving, and thus increased awareness, had a positive lessening effect on the energy consumption of residents. The study, conducted in South Korea, found that collective efforts created community spirit, which can be another motivating factor for residents to continue these efforts.

Looking at human factors related to energy usage in residential buildings [6], researchers found that for most of the people in the studied cohort comfort meant warmth, but that actions giving most comfort were not energy intensive. Saving money was found to be the greatest motivator for changing energy consuming behaviors.

We have focused on IoT data visualization in an earlier study [8] where we presented methods of utilizing the free map services available on the Internet for the visualization of the gathered IoT data. In the system under study, monitoring of road conditions was crowdsourced. Conditions were monitored by drivers using an Android application. This application sends the timestamped data about acceleration, speed, location, as well as other data to the server. OpenStreetMap and Google Maps were used to visualize the routes driven. The condition of the roads calculated from the data was indicated by the color of the different sections of the path.

In addition, the appropriate tools for storing, monitoring, and visualizing the data of living conditions have been presented [9]. In the context of real world IoT systems different types of databases were tested for storing and retrieving data: time series, document, and relational databases were compared. For visualization of the IoT data open source Grafana visualization and analytics web application was used.

III. Architecture

The architecture of DataSites system is shown in Fig. 1.

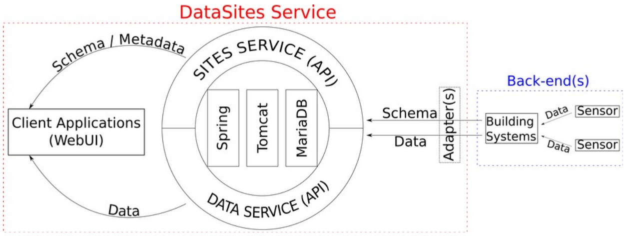

Figure 1: Architecture of DataSites service

The five architectural components comprising DataSites system are:

1. Building systems
2. Adapters
3. Data and Sites service components
4. Data and Sites services' APIs and Data Formats
5. Client applications

Building systems are the building automation systems and sensors from which DataSites can retrieve or receive data. Communicating with building systems DataSites' is the responsibility of adapter components. The information streaming from automation systems and sensors is in heterogeneous data formats, so DataSites' adapters also need to convert the data to JSON object arrays, which include the data most pertinent to users.

Two categories of data are handled by DataSites:

- the physical layouts of buildings, called *schemas* in DataSites, and
- the measurements and other data sent by the building systems.

Both of these data categories require using adapters to convert them. Sites service and its API are used to manage the data concerning the physical layouts of the buildings, and Data API is responsible for managing other data. APIs are REST (REpresentational State Transfer) APIs described with OpenAPI specifications. Clients use HTTP protocol to send requests to API servers to interact with their resources. An effort was given to keep the APIs simple to ease client application development.

IV. Requirements for visualizing collected building system data

Visualization of data involves the presentation data in the graphical form, as graphs, charts, histograms, gauges, geographic maps, etc. To be useful to its users, the visualizations of building data have to be able to show the data in understandable form, and in its proper context. Here the context is a building site, which vary in their layout complexity. A larger building site can have several multi-floor buildings on them, with several rooms and sensors. Using DataSites' data models for sites, layers, and sensors users can place a sensor in a room on the third floor of a building. Layers can be used to model the floors and rooms, as well as other spaces of a building. Used in this manner, the visualization provides an intuitive way for the end-users to connect the source of the sensor data to the real world.

The building systems change during the lifetime of the building. There will likely be changes to the sensors attached to the building system. Sensors might need to be replaced by newer or different models, or be moved if their initial installation spots were suboptimal. Placing the sensors in their proper places in layers is made easier by the fact that DataSites' sensor data model has x, y, and z attributes for three-dimensional coordinates. This way a sensor's location in the visualization is accurate.

Building systems, sensors and other related IoT systems create a voluminous amount of data which places requirements for the servers, and clients too. As some sensors broadcast their measurements several times per minute, or even in intervals of a couple of seconds, these measurements create a discrete time series of measurements. Storing this data requires hard disk space, and processing larger time series data sets demands high computing power from clients. DataSites uses the InfluxDB time series database (TSDB) [11] to store data. Published studies [12-14] have shown that InfluxDB has advantages over many of the other options in performance, especially in executing aggregation functions over data and faster grouping of query results.

Stored time series data is intuitive to visualize and understand. Graphs can be drawn using time as the x-axis, with y-axis reflecting the changes in measurement values. Errors in time series, like erroneous values caused by sensor malfunction or periods where the sensor didn't send data, can be easy to spot from a timeline graph. An example that can be seen from a timeline series, is sensor drift. In the sensor drift the sensor's ability to send reliable readings deteriorates over time.

One use of building system data is monitoring the proper functioning of buildings. In case of detected problems, manifesting as abnormally high or low sensor measurements, most building systems are configured to send alerts to the appropriate personnel. These abnormal readings are usually highlighted graphically in visualizations, for example using colors, larger text, or opening a dialog showing a warning. In buildings of course a wide range of problems can arise: pipes start to leak, electric fuses operate, or the heating system goes offline, etc. These problems need to be shown in a clear way to the users in the visualization too. Serious problems should be placed at the central place to alert the users. If a fire alarm or a burglary alarm system is integrated into the visualizations, showing active alarms from them is more important than showing other information.

V. SITES SERVICE AND WEBUI

This chapter first discusses the Sites Service and its API. Then it examines DataSites' WebUI visualization tool through an example of creating a visualization of sensors on a building site to illustrate this process.

Data from both the Sites Service's and Data Service's OpenAPI REST APIs is utilized by WebUI client in visualizations. The Sites Service's API provides access to three types of resources: sites, layers, and sensors. These all have associated endpoints in the API. The creation, the modification, and the deletion of resources is enabled through these endpoints.

In the communication between the clients and the service API, resources are transmitted as representations in JSON format. The data models for these representations are specified with OpenAPI with the endpoints. All data models have a required ID field. The data model for the sites also requires a name attribute. For the layer's data model, the following attributes are required: index for ordering the stacked layers, the layer name, and the type of the layer. Layer can be one of five types: unknown, outdoor, a building, a floor, or a room. This set of types were chosen to give enough flexibility in specifying the layers, while keeping the set down to a manageable size. Mandatory fields for a sensor data type are an external ID, and the sensor's service type. External ID is the ID of the sensor in the building system the sensor is sending its data to. Service type is the name of a specific building system. DataSites needs to be configured to what type of data is available from a building system. After that DataSites can use the external ID and service type to fetch a sensor's data.

Figure 2: Creating a building site

Before creation of any visualization for a new site, certain resources and information are required. End-users' requirements for the visualization drive the process, so gathering them is the first step. To meet the requirements, developers need to have adequate information about the site and all the building systems to be visualized. Access rights to the data from building systems and sensors on the site have to be obtained from the site's owner or manager. Communication involved in this can take time, as often it is the vendor of the building system, not the site manager or owner, who actually can enable access to the data. Images indicating the physical layout of the building (floor plans, blueprints, etc.) can be added to layers like floors of the building, or rooms, and need to be at hand, too.

The creation of the site visualization starts with creating a new site in WebUI, as shown in Fig. 2. In the dialog named "Add a site", only the name of the site needs to be given, but it is good practice to also include a description for the building, and as well as its managing organization. A web URL for the site can be given if one exists. Here we create a site named "Amazing building". Once created the site will be available from the dropdown menu on the uppermost-left corner.

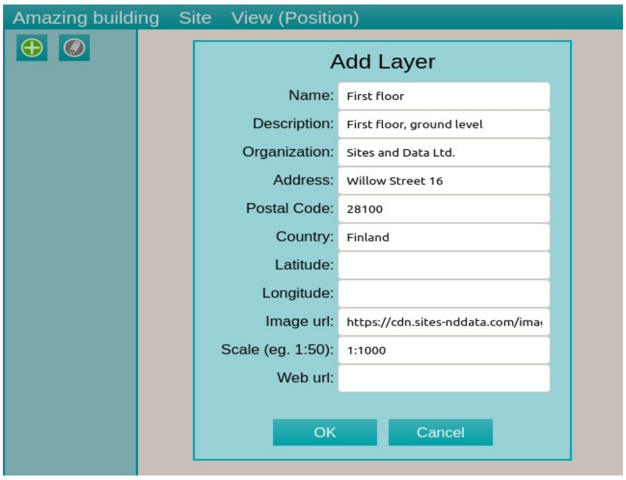

Figure 3: Adding a layer to a site

We can now add a layer to the new site, in this case the first floor of the building. In "Add a layer" dialog (see Fig. 3) we can enter information for the layer. Here again we have the option of providing not only the mandatory name attribute, but other available information like the postal address connected with the layer. One attribute of note is the "Image url", which is used to link to an image file picturing the floor plan or similar. This image is

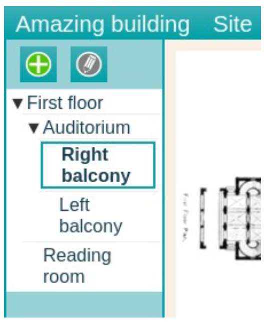

Figure 4: Hierarchy of layers in a site

always shown when the layer is selected from the side panel. The link can be to any image file hosted on a web server DataSites can access.

The created layer is viewable on the left side panel of WebUI. When we choose the layer from that panel, we can create other layers as its children. All layers can have children. Hierarchies can be used to model complex buildings, like one in Fig. 4.

Adding sensors starts with selecting the layer for which the sensor will be bound. A layer's visibility dictates if the sensor is shown in the UI, a sensor will only be visible if the layer it is on has been selected from the hierarchy on the left side panel. Dialog "Add sensor", like in Fig. 5, requires the name, external ID, and service type. As was stated earlier, external ID enables accessing the sensor's data in the remote building system, and the service type is used to deduce what information is available from this building system.

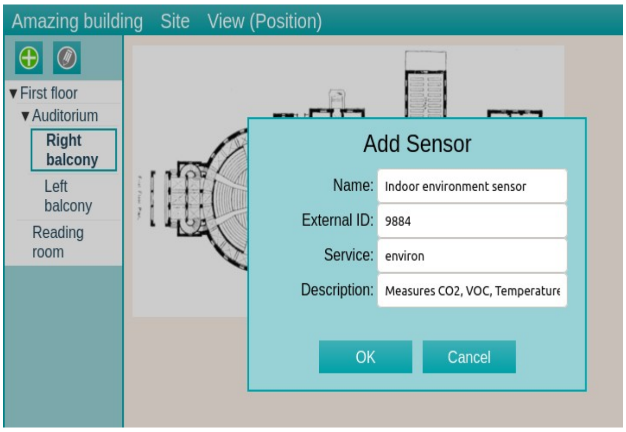

Figure 5: Adding a sensor to a layer

A turquoise square is created as the sensor icon. This icon can be moved on the layer to place it on the exact location of the sensor. Clicking or hovering over the icon opens a window showing the latest measurements received from the sensor. Fig. 6 shows an example of a sensor whose service type is setup to provide readings for temperature and target temperature, as well as values for valve, signal, and charge level.

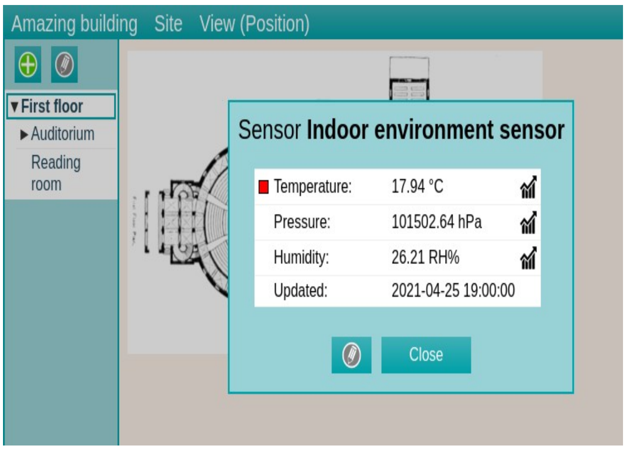

Figure 6: Window showing the latest readings from a sensor

By default, only the icon for the sensor is shown in its location. However, this can be changed from the top bar selection "Show". In the menu that opens, all the measurements available from the sensor in DataSites are visible. If we chose for example "Temperature", in the same location we would see the real-time temperature readings from the sensors that send that data.

WebUI enables users to set any number of alert states for each measurement. By setting the minimum and maximum values for each alert state of the measurement, the sensor sending that abnormal value causing the alert is highlighted with the background color the user chooses.

The process of adding layers to the site, and sensor to those layers is repeated until the desired parts of the site have been modeled.

DataSites can also visualize history data of a quality measured by a sensor. Fig. 7 shows a graph visualization of temperature time series data as measured by a sensor. Time period visualized has been chosen to be 14 days, other options are 1, 5, or 30 days. Chart.js [14] JavaScript library is used for this visualization.

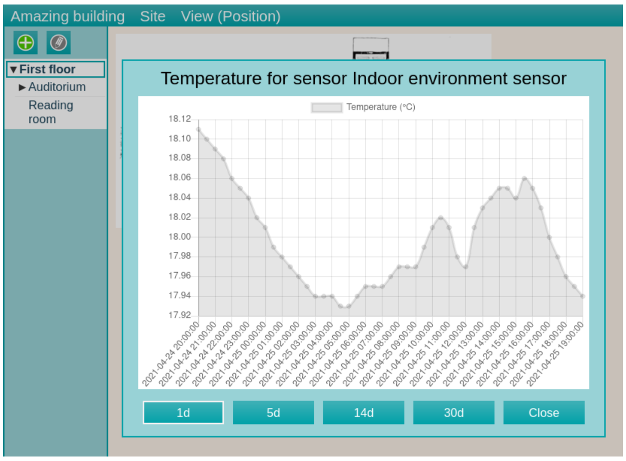

Figure 7: Graph visualizing the history data of room temperature measurement

WebUI client application was developed using JavaScript, HTML5, and CSS3. jQuery and Underscore JavaScript libraries were used. The application was served from the same Java Spring Boot as the Site Service itself.

VI. DISCUSSION AND CONCLUSION

The study is part of our ongoing project, the purpose of which is to develop a means of reducing energy consumption in houses and apartments.

The presented DataSites service and its WebUI enable users of buildings to get real-time data of living conditions. WebUI, and tools like it, that have been designed to increase a building's users' awareness of the living conditions. Maintaining comfortable living conditions has direct links to energy consumption, for example the temperature difference between indoors and outdoors implies that energy has been used for cooling or heating.

As the goal of the KIEMI project is to reach the minimal energy consumption while maintaining comfortable living conditions, it would be useful to integrate direct energy consumption readings into the visualizations. Also, the correlation of changes in the living conditions and in energy consumption could then be visualized. When users are able to discover these correlations, and what behaviors affect energy usage most, could give them incentive to change these behaviors.

The cost of the energy should be made clear in the visualizations, too. This is supported by one of the key findings from a study [6]: saving money was the most important motivating driver in energy-saving behavior. This would expand the systems DataSites needs to communicate with to include energy systems and smart meters, and systems from which real-time electricity prices can be fetched.

One important aspect of DataSites is implementing adapters that ensure that the data visualization components receive are in unified format. This has proven to be a time-consuming part of the process, as the adapters have had to be created for each building system. Now the vendors of building systems are hesitant to open and document their interfaces.

For residential buildings, DataSites could be developed to act as a Home Energy Management System (HEMS), which could include dedicated In-Home Displays (IHDs). Smart meters, capable of capturing the electricity consumption of the devices at home can give high definition data which can be stored and visualized. However, a study looking at HEMS [14] use found that their effect on energy usage was highly dependent on the household they were installed in, some reducing energy usage, but others actually increasing it. Also, a study [16] found that the aesthetics of IHDs affected the frequency users used it. This has implications to UI design too. Graph.js and other visualization libraries could be used to create visually more interesting graphical representations of the building system data.

It would be interesting to create energy saving feedback systems, maybe with DataSites, that would be able to create community spirit around this matter, like in previous research [5]. This will require future work on how the social dynamics can be affected with these systems. Crowdsourcing energy-saving measures could be one direction worth exploring, similarly to what was done with a road maintenance system we studied before [8].

VII. Summary

This paper presented a user interface for providing relevant data for the users, managers, and owners of buildings. Furthermore, requirements for visualizing building data were described. It was discovered that to be useful to its users, the visualizations of building data have to be able to show the data in understandable form, and in its proper context. To achieve this goal, this paper also introduced asimple layer-based data model for describing building features, such as floors, rooms and sensor locations. The developed WebUI showed promise, and we will continue the development of the service in our ongoing project.

Acknowledgment

This work is part of the KIEMI project and has been funded by the European Regional Development Fund and the Regional Council of Satakunta.

References

[1] Official Statistics of Finland (OSF): Energy supply and consumption [e-publication].

[2] L. Bartram, J. Rodgers and K. Muise, “Chasing the Negawatt: Visualization for Sustainable Living,” in IEEE Computer Graphics and Applications, vol. 30, no. 3, pp. 8-14, 2010.

[3] M. Saari, P. Sillberg, J. Grönman, P. Rantanen, H. Jaakkola, and J. Henno, “Survey of Applications for Apartment Energy Consumption Monitoring,” in 23rd IEEE International Conference on Intelligent Engineering Systems 2019, IEEE, 2019.

[4] J. E. Petersen, V. Shunturov, K. Janda, G. Platt, and K. Weinberger, “Dormitory residents reduce electricity consumption when exposed to real-time visual feedback and incentives,” International Journal of Sustainability in Higher Education, Vol. 8, No. 1, pp. 16-33, 2007.

[5] N. N. Kang, S. H. Cho, and J. T. Kim, “The energy-saving effects of apartment residents’ awareness and behavior,” Energy and Buildings, Vol. 46, pp. 112-122, 2012.

[6] G. M. Huebner, J. Cooper, and K. Jones, “Domestic energy consumption - What role do comfort, habit, and knowledge about the heating system play?” Energy and Buildings, Vol. 66, pp. 626-636, 2013.

[7] A. Faruqui, S. Sergici, and A. Sharif, “The impact of informational feedback on energy consumption - A survey of the experimental evidence,” Energy, Vol. 35, No. 4, pp. 1598-1608, 2010.

[8] P. Sillberg, M. Saari, J. Grönman, P. Rantanen, and M. Kuusisto, “Interpretation, Modeling, and Visualization of Crowdsourced Road Condition Data,” in Intelligent Systems: Theory, Research and Innovation in Applications, R. Goncalves, V. Sgurev, V. Jotsov, and J. Kacpzyk, Eds. Springer, pp. 99–119, 2020.

[9] M. Saari, J. Grönman, J. Soini, P. Rantanen, and T. Mäkinen, “Experimenting with Means to Store and Monitor IoT based Measurement Results for Energy Saving,” in 2020 43th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2020.

[10] M. Nurminen, P. Rantanen, M. Saari “DataSites: a Simple Solution for Providing Building Data to Client Devices”, in 2021 44th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2021. Submitted.

[11] InfluxDB 2.0 documentation, https://docs.influxdata.com/influxdb/v2.0/, Retrieved: January 29, 2021.

[12] S. Di Martino, L. Fiadone, A. Peron, V. N. Vitale, and A. Riccabone, “Industrial Internet of Things: Persistence for Time Series with NoSQL Databases,” Proceedings -2019 IEEE 28th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises, WETICE 2019, pp. 340–345, 2019.

[13] E. Musa, G. Delač, M. Šilić and K. Vladimir, “Comparison of Relational and Time-Series Databases for Real-Time Massive datasets,” in 2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2019, pp. 1065–1070.

[14] Chart.js 2.9.4 documentation, https://www.chartjs.org/docs/2.9.4/, Retrieved: February 4, 2021.

[15] A. Nilsson, M. Wester, D. Lazarevic, and N. Brandt, “Smart homes, home energy management systems and real-time feedback: Lessons for influencing household energy consumption from a Swedish field study,” Energy and Buildings, Volume 179, 2018,

[16] T. Hargreaves, M. Nye, J. Burgess, “Making energy visible: A qualitative field study of how householders interact with feedback from smart energy monitors”, Energy Policy, Volume 38, Issue 10, 2010.