Demo Abstract: Vista: Spatial Data Representation for Smart Buildings

Matilda Kathryn Ferguson
Swarthmore College, Swarthmore
mfergus3@swarthmore.edu

Sudershan Boovaraghavan
Carnegie Mellon University, Pittsburgh
sudershan@cmu.edu

Yuvraj Agarwal
Carnegie Mellon University, Pittsburgh
yuvraj@cs.cmu.edu

ABSTRACT

With the increasing prevalence and power of building IoT sensors and devices, there is a growing need for intuitive, accessible, and meaningful data visualization tools that are specific and tailored for building IoT. Existing tools that incorporate spatial contextualization are insulated to specific applications with predefined visualization goals. We present Vista, a front end tool, that presents dynamic data visualizations of building IoT data on an interface that allows for context and deeper evaluation of information. This framework proposes a method to transform any static building floor plan into a dynamic one with which one can create unique and intuitive data representations tailored to their own objectives. The modular design of this tool enables an authoring system flexible to heterogeneous data by separating the visual elements of the floor plan and the relevant data, enabling diverse visual presentations of various types of IoT information and inputs.

CCS CONCEPTS

• **Human-centered computing → Visualization toolkits**.

KEYWORDS

Data Visualization; Visual Inference; Smart Buildings;

ACM Reference Format:
Matilda Kathryn Ferguson, Sudershan Boovaraghavan, and Yuvraj Agarwal. 2020. Demo Abstract: Vista: Spatial Data Representation for Smart Buildings. In *The 7th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys '20), November 18–20, 2020, Virtual Event, Japan.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3408308.3431112

1 INTRODUCTION

The vision of smart buildings is enabled, and in many cases even accelerated, by the availability of a wide variety of novel IoT sensors [6], expected to reach Billions of IoT devices by 2020. Powered by this multi-modal sensor data, building managers can control and optimize the operation of their buildings, making them more efficient and performative in order to meet the needs of occupants. Occupants can also be provided data about their own personal spaces, or shared common areas for improving comfort, providing building use context, and for reporting any issues. Despite this growing deluge of IoT powered data in buildings, the state of the art to *display or visualize* this data is typically plotting time series of sensor data [4, 8]. We believe that this rich IoT sensor data is immensely more valuable when augmented with additional location and spatial context, such as displaying it *live* on a floor plan. For example, rather than just viewing time series of the temperature of an office, a manager can view the temperature variation overlaid on the entire floor plan of a building to detect spatially co-located rooms with issues due to a problematic Variable Air Volume (VAV).

Recognizing this need to provide sensor data with spatial contexts, there are efforts to build dashboards with floor plans [2, 5] with sensor data. However these efforts are often custom jobs requiring time consuming plan creation by companies, manual connections to data sources that are not scalable. We designed Vista, an intuitive, robust, and modular data visualization system for building scale IoT data with the overarching goal of providing IoT data in its appropriate spatial context. At its core, Vista has three components. The first is a methodical way to convert existing 2D floor plans in common formats (e.g. PDF, DXF, DWG) into a flexible Scalable Vector Graphic (SVG) format with anchors to identify physical spaces and objects to overlay. Second, a module with different data visualization modalities such as icon overlays, heat maps and time series. Finally, a flexible and extensible way to connect IoT data sources using a variety of standard methods such as REST/PubSub APIs [8] or even historical data files, and overlay them using the defined anchors in a customizable manner. We have used Vista to create dynamic spatial maps for multiple floors of two buildings on our campus. It takes 6-7 hours to create this overlay for an 100,000 sq-ft building with four floors. We also provide a case study where Vista helped debug issues in the WiFi network already.

2 SYSTEM OVERVIEW

We designed and built Vista as a programmable overlay of architecture floor plans. In the design of Vista, we particularly emphasized *flexibility* by ensuring users have freedom in the type of data they visualize and how it is presented. We bridge the gap between data collection and visualization through anchored visual objects overlaid on a traditional floor plan. We then simplify the data as part of Vista to visualize a wide variety of IoT sensor data. The interface is geared towards building managers and occupants of the building rather than an architect. With Vista, we were able to create a comprehensive visualization UI that presented dynamic and meaningful presentations of building data built with d3.js [3]. We combine traditional time series data presentation with innovative spatial representation to present users with a comprehensive understanding of the data and conditions that they care about.

BuildSys '20, November 18–20, 2020, Virtual Event, Japan

ACM ISBN 978-1-4503-8061-4/20/11.
https://doi.org/10.1145/3408308.3431112

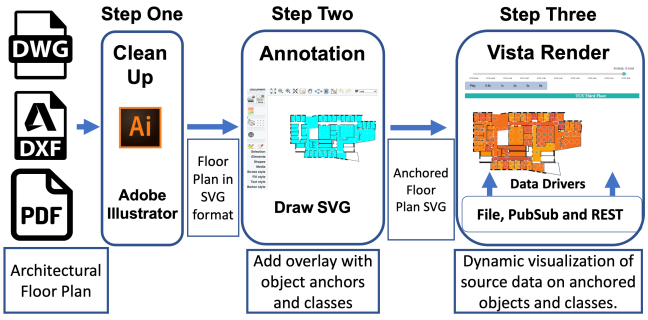

Figure 1: A Vista workflow. A user starts from an architectural drawing, simplifies it to remove unnecessary detail, and then adds annotated overlays depicting physical spaces (e.g. rooms) as well as objects anchors (e.g. location of a sensor). These are used by our Vista renderer to overlay real-time and historical sensor data on top of it.

2.1 Design and Implementation

Architectural plans are designed with extraneous details that are not useful in the building IoT context - including textual information, architectural cues, and markings of building details such as window openings. We used the graphic design tool Adobe Illustrator [1] to remove these elements and present a more intuitive and *cleaner* presentation of the building. Using this graphic as a base background image, we used an online software DrawSVG [7] to create an annotated and programmable overlay, essentially tracing over the simplified floor plan in the background. Each room element is traced, anchored, and given a unique id. We developed and implemented a uniform naming convention - whereby objects were identifiable by room number, floor, spatial placement, and in the case of IoT sensors, a unique id corresponding to the IP address (or the MAC address) of that particular sensor. Visual objects were grouped by class, for example *room* or *sensor*, in order to maximize utility and association. To ensure the correct placement of sensor anchors on the overlay, a marked sensor plan is temporarily overlaid to create a reference for sensor placement and subsequently removed. The online software exports the annotated floor plan in the SVG format, which can be downloaded, scaled, and placed directly into the UI.

The manually added anchored elements within the floor plan are visually adapted through predefined references, making it intuitive to create novel data representations. The unique id associated with visual elements allows individual mapping of the sensor IDs in Vista with their actual data sources in a live time series database, or in our case the BuildingDepot [8], a smart building OS, using REST and PubSub APIs. BuildingDepot uses the notion of tags that allow sensors to be connected with their anchor points in the SVG floor plan by using associated IDs. Anchored elements become associated with tagged metadata of sensors and rooms within the API, such as room dimensions. Vista dynamically updates the corresponding anchor points with the data stream for the entire floorplan.

Vista includes a tool that allows synced time-series data to be skimmed through, such that users can view data patterns as necessary. Since this is synced with varying anchored elements, this tool can be applied to manipulate the sensor elements directly or to make heat maps for entire rooms on the scale of the building.

2.2 Use Cases

We present two use cases of our front-end tool in real world application that was deployed and evaluated. We used Vista as a *debugging tool* to spatially represent where reboots for devices deployed on our campus were occurring every hour for several days. The floor plan was anchored with the sensor location and displayed with a surrounding circle, which was color filled with a corresponding scaled RGB value for the number of reboots. A slider tool allowed the building administrator to skim through data and locate problematic sensors and hot spots of poor network connection. The administrator provided a testimonial that Vista was incredibly useful to them in finding and fixing the sensors which were having issues connecting to the network, more so than the simple identification of problematic sensors by IP addresses. Another application of this tool is the visualization of *live data*. We linked visual elements to a subscribed sensor data stream, enabling the visualization to constantly update the fill of sensor elements with color scaled values of data points while the user remained on the page. Applied to the entire building, users have the capability to notice data abnormalities immediately.

3 CONCLUSION AND FUTURE WORK

Modern user interface tools fail to incorporate effective spatial contextualization of IoT data and are built for specific purposes. We present Vista, a front-end user interface that produces a dynamic visualization of IoT data spatially over floor plans of a building. We used Vista to create dynamic floor plans of two buildings on our campus each spanning 100,000 sqft and demonstrate the visualization data for the sensors deployed in the building. Vista's process of creating visualizations needs to be streamlined such that it can handle unanticipated data formats with minimal updates to the programming. We plan to test the efficacy of Vista via a user study that examines user benefits and work speed differences.

This research has been supported in part by NSF awards TWC-1564009 and SaTC-1801472.

REFERENCES

[1] Adobe Inc. [n.d.]. *Adobe Illustrator.* https://adobe.com/products/illustrator
[2] CubiCasa. 2020. *CubiCasa | Create a floor plan in under 5 minutes.* Retrieved September 21, 2020 from https://www.cubi.casa/floor-plans-smart-homes-iot/
[3] D3.js. 2020. *D3.js - Data-Driven Documents.* Retrieved September 21, 2020 from https://d3js.org/
[4] Gabriel Fierro and David E Culler. 2015. Xbos: An extensible building operating system. In *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments.* 119–120.
[5] FloorPlanMapper. 2020. *HOME - Floor Plan Mapper - Interactive Office Floor Plan Mapping.* Retrieved September 21, 2020 from http://www.floorplanmapper.com/
[6] Gierad Laput, Yang Zhang, and Chris Harrison. 2017. Synthetic sensors: Towards general-purpose sensing. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems.* 3986–3999.
[7] Joseph LIARD. 2020. *DRAWSVG The free online drawing tool.* Retrieved September 21, 2020 from https://www.drawsvg.org/
[8] Thomas Weng, Anthony Nwokafor, and Yuvraj Agarwal. 2013. Buildingdepot 2.0: An integrated management system for building analysis and control. In *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings.* 1–8.