A Demonstration of PyViz, a Flexible Smart Home Visualization Tool

Brian L. Thomas
School of Electrical and Computer Science
Washington State University
Pullman, Washington 99164–2752, USA
Email: brian.thomas@email.wsu.edu

Aaron S. Crandall
School of Electrical and Computer Science
Washington State University
Pullman, Washington 99164–2752, USA
Email: acrandal@wsu.edu

***Abstract*—As smart home technologies continue to be deployed in research and real world environments, there continues to be a need for quality visualization of the smart home data. This data come from a variety of sensor sources, artificial intelligence algorithms and human feedback. Building tools that are easily configured, rapidly developed and capable of communicating with both real time and historical data is ever challenging.**

This work introduces PyViz, a smart home focused middleware interactive interface. PyViz is the result of several years of testing and development in a large smart home research project. During its development, issues of configuration, ease of use and adaptability have been addressed. The demonstration of this tool hopes to enlighten the audience about the facets of user interfaces, network agents and smart home complexity.

I. Introduction

With the advent of modern smart home technologies, the ever increasing complexity of the data available makes it difficult for humans to understand the current workings of the systems around them. These new smart homes have been developed to support green technologies, medical applications and aging in place methods, among others. The wide variety and volume of data produced by the sensor platforms need to be made available in a format that the human users of the system may consume and understand to bring these technologies to their full potential.

To accomplish this, a large number of rendering and visualization tools have been built. These tools have ranged from simple 2D renderings of the system state [1], statistical histories of single sensors [2], [3] to complex 3D renderings of the full smart home space [4]. All of the visualizers created for smart home applications have benefits and limitations depending upon their goals, but constructing a simple and adaptable tool for rendering smart homes and their sensors requires a good balance of programming tools, data formats and graphical rendering.

During the evolution of the Center for Advanced Studies in Adaptive Systems (CASAS) at Washington State University smart home research project [5], a suite of visualizers have been developed and evaluated. These have ranged from statistical mappings of the sensor state to complex 3D models. Of the tools constructed, PyViz has been found to have the most capable balance of features and adaptability. This visualizer leverages the Python programming language and standards-based configuration formations to provide a feature-rich, rapidly deployable method of viewing the smart home system in real-time (Fig. 1) as well as provide access to historical trends (Fig. 2).

Since its inception, having a platform that lends itself to rapid development has allowed PyViz to be a keystone of the CASAS smart home research. It is now used during all stages of smart home deployment, including planning, installation, live operation and historical assessment. Part of its flexibility has been its ability to draw from multiple data sources, including the CASAS Lightweight Middleware (CLM) [6], SQL databases and local files. Additionally, as new artificial intelligence algorithms are developed to model the smart home, such as Activities of Daily Living detection [7], resident tracking [8] or behavior modeling [9], [10], the current state of those models can quickly be rendered to the PyViz interface. Having such a versatile tool available has been invaluable to the continued research and deployment of smart home technologies.

II. Demonstration

The proposed demonstration of this tool is intended to highlight and contrast the capabilities of the PyViz tool by addressing:

- Who benefits from visualization of these smart home environments.
- What aspects of the smart home data are most important to the researchers and residents.
- When a visualizer can make a serious difference in a research project.
- Where these tools need to be deployed, during both the research phases of a smart home and for residents to be shown the results of these systems in their homes.
- Why visualization is important instead of just statistics or logged output.
- How the CASAS project proceeded to design and evaluate these tools for their needs.

The goal is to give the attendees a strong sense of how combining physical sensors in complex environments with the right software makes a difference to their project results. Smart homes are a diverse fusion of sensors, artificial intelligence algorithms, computation, environmental controls and people.

Fig. 1: The PyViz tool rendering a smart home installation in an office space, codename "Tokyo".

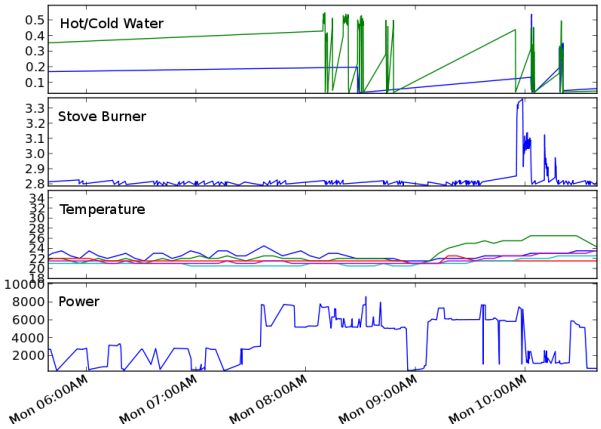

Fig. 2: PyViz graphing historical numeric data from the CASAS smart home enabled apartment.

This demonstration should show why making good selections in standards, interfaces and middleware assists in the development of these systems.

A. Demonstrated Application

The primary application demonstrated is PyViz, a smart home focused data visualizer. The presentation will go through the various stages of using the tool, from building a smart home configuration file in a Scalable Vector Graphics (SVG) format [11], to watching a sample environment live, to reviewing historical data and finally the integration of other artificial intelligence tools, such as a resident tracker. This range of features will be set in to full scale smart home environments, (shown in Fig. 1 and 3) as well as a small live sensor platform.

The ability for PyViz to interact with the CASAS middleware as an agent in real time will be discussed. Additionally, connecting to a SQL database or using text files as data sources will be introduced. These features are essential for a user interface to be deployable both for live viewing and historical analysis of the smart home behavior.

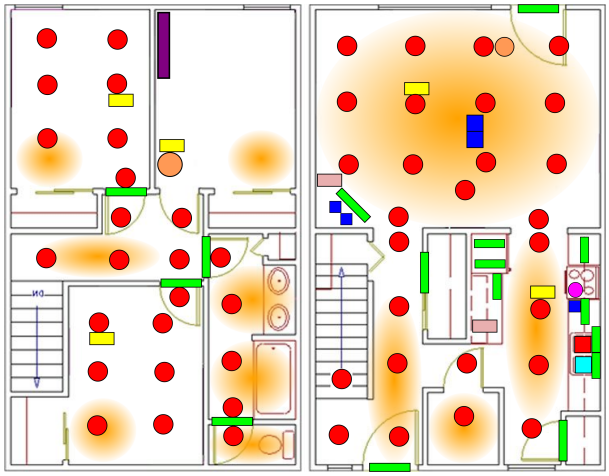

Fig. 3: Graphically rendered SVG configuration file for our site with codename "Kyoto".

Lastly, the benefits and negatives of using an interpreted language such as Python will be included. This language provides rapid prototyping and ease of design, but introduced limitations when dealing with large volumes of historical data. An example of the benefits of using such a language will be shown in the form of adding a means to visualize a new form of data coming through the middleware. In this case, it will be the inclusion of output from an AI modeler that is tracking the residents as they move about the smart home. This new feature was included quickly, giving the researchers rapid feedback about the capabilities of their new algorithm.

B. Sample Hardware

This demonstration will utilize a set of sample sensors integrated onto a demonstration sensor board. A SheevaPlug^{TM}small form factor computer will operate as the center of the simple smart home system. Our presentation laptop and the server will connect to an attached Ethernet router for network communication. Attendees will be able to interact with the sensors (motion detectors, miniature doors, temperature, light, shake sensors, etc.) and see their interactions visualized in real time, or change the configuration file and see how the changes effect PyViz.

C. Demonstration Steps

Creating a new site configuration: The first step of our demonstration shows the benefits of using a standardized tool for configuring a new site. We will go through the steps of creating a simple site configuration for our sensor demo board. After configuring an adequate number of sensors the presentation will proceed to the next step in the demonstration.

Visualizing live data: We will load the completed site configuration for our sensor demo board into PyViz and connect to the CLM communicating with our sensor demo board. This will allow us to render our interaction with the sensors in real time before the audience.

Visualizing historical data: Now that we have shown PyViz rendering live data from the CLM, it is time to demonstrate some of the other features a specialized visualizer can provide A local data file from the CASAS archives will be loaded into PyViz to show some of the features you can use when playing back data, including historical trends. Connecting to and pulling from our archival database would be impossible as it is limited to local network connections with authentication in our lab for security reasons, so we will just mention this feature.

Adaptability for additional smart home uses: In the final portion of our demonstration a video of PyViz rendering live data from our lab will be shown. It will show additional information from one of our smart home modeling algorithms attempting to track individuals moving through the space. This demonstrates just one of the many possibilities of information visualization that can be accomplished with the rapid and adaptable prototyping we can accomplish with PyViz.

III. Technical Requirements

This demonstration will require the use of a projector with a VGA connection, and 2 3-prong wall outlets to power our laptop and portable smart home equipment.

Acknowledgments

This work is supported by NSF grant IIS-0121297.

References

[1] Y. Ivanov, C. Wren, A. Sorokin, and I. Kaur, "Visualizing the history of living spaces," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1153–1160, November 2007.

[2] W. Müller and H. Schumann, "Visualization methods for time-dependent data - an overview," in *Proceedings of the Winter Simulation Conference*, S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice, Eds., vol. 1, December 2003, pp. 737–745.

[3] M. Rantz, M. Aud, G. Alexander, D. Oliver, D. Minner, M. Skubic, J. Keller, Z. He, M. Popescu, G. Demiris, and S. Miller, "Tiger place: An innovative educational and research environment," AAAI Fall Symposium, Washington, DC, Tech. Rep. FS-08-02, November 2008.

[4] S. Szewcyzk, K. Dwan, B. Minor, B. Swedlove, and D. Cook, "Annotating smart environment sensor data for activity learning," *Technology and Health Care*, vol. 17, no. 3, pp. 161–169, August 2009.

[5] WSU CASAS, "Center for Advanced Studies in Adapative Systems (CASAS) at Washington State University," http://casas.wsu.edu/, 2010.

[6] J. Kusznir, "CLM as a smart home middleware," Master's thesis, Washington State University, 2009.

[7] G. Singla, D. J. Cook, and M. Schmitter-Edgecombe, "Recognizing independent and joint activities among multiple residents in smart environments," *Journal of Ambient Intelligence and Humanized Computing*, vol. 1, no. 1, pp. 57–63, 2010.

[8] A. S. Crandall and D. J. Cook, "Tracking systems for multiple smart home residents," in *Human Behavior Recognition Technologies*. IGI Global, 2010.

[9] R. Aipperspach, E. Cohen, and J. Canny, "Modeling human behavior from simple sensors in the home," in *Pervasive Computing*, ser. Lecture Notes in Computer Science, K. Fishkin, B. Schiele, P. Nixon, and A. Quigley, Eds. Springer Berlin / Heidelberg, 2006, vol. 3968, pp. 337–348.

[10] I. J. Sledge, J. M. Keller, T. C. Havens, G. L. Alexander, and M. Skubic, "Temporal activity analysis," AAAI Fall Symposium, Washington, DC, Tech. Rep. FS08-02-016, November 2008.

[11] J. Ferraiolo, D. Jackson, F. Jun, A. Grasso, D. Jackson, C. Lilley, C. McCormack, D. Schepers, J. Watt, and P. Dengler, "Scalable vector graphics (SVG) 1.1 specification," W3C, W3C Recommendation, January 2003. [Online]. Available: http://www.w3.org/TR/2003/REC-SVG11-20030114/