Modelled Testbeds: Visualizing and Augmenting Physical Testbeds with Virtual Resources

Stephane Kundig[1], Constantinos Marios Angelopoulos[2], and Jose Rolim[1]

[1] University of Geneva, {stephane.kundig,jose.rolim}@unige.ch
[2] Bournemouth University, mangelopoulos@bournemouth.ac.uk

Abstract. Testbed facilities play a major role in the study and evolution of emerging technologies, such as those related to the Internet of Things. In this work we introduce the concept of modelled testbeds, which are 3D interactive representations of physical testbeds where the addition of virtual resources mimicking the physical ones is made possible thanks to back-end infrastructure. We present the architecture of the Syndesi testbed, deployed at the premises of University of Geneva, which was used for the prototype modelled testbed. We investigate several extrapolation techniques towards realistic value assignment for virtual sensor measurements. K-fold cross validation is performed in a dataset comprising of nearly 300'000 measurements of temperature, illuminance and humidity sensors collected from the physical sensors of the Syndesi testbed, in order to evaluate the accuracy of the methods. We obtain strong results including Mean Absolute Percentage Error (MAPE) levels below 7%.

Keywords: internet of things; testbeds; modelling; visualization;

1 Introduction

In the fields of computing and networking, experimental facilities and testbeds play a key role in studying and evaluating new technologies. They provide the necessary controlled environment and tools that enable researchers to run experiments and evaluate novel protocols and architectures. They support agile architectures that are easy to re-configure in the context of experiments and provide additional services and tools for collecting meta-information on the experiment execution (e.g. monitoring several performance metrics or providing execution logs for post-experiment processing).

However, despite the services and the advantages provided, testbeds also pose some limitations. By nature, each testbed facility focuses on a specific area of interest (e.g. in Internet of Things (IoT) applications or Machine-to-Machine (M2M) communication protocols) and therefore its architecture and the services provided define the supported experiments. Another limiting factor is the number of available resources and subsequently the number of experiments that can be run simultaneously. In general, scaling up an experimental facility either for increasing its size, the number of simultaneous experiments supported or for improving its availability, infers high costs and requires significant effort.

Trying to address the aforementioned restrictions of traditional testbed facilities in IoT research, we introduce a new type of facilities; namely *Modelled Testbeds*. A modelled testbed consists of two components. The first component consists of physical IoT resources (e.g. IPv6-enabled sensor motes) that are actually deployed in the premises of the facility. The resources operate with the aid of a back-end system that orchestrates and monitors their operation, similarly to any other regular testbed. The second component consists of virtual resources that quantitatively and qualitatively augment the physical component of the testbed. In particular, a digital model of the physical component is extracted capturing the space of the facility and the operation of the physical resources. Then, via an intuitive Graphical User Interface (GUI), virtual resources can be spawned and deployed within the digital model in positions that correspond to the actual premises of the facility. The operation of the virtual resources is simulated in a way that captures the characteristics of the physical space (e.g. light distribution) by drawing information out of the physical sensors. This way the simulation of the virtual resources is seeded by the physical component and, therefore, their operation is intertwined with that of the physical resources. The end result is a mixed set of physical and virtual resources whose operation is transparently perceived by the end-user as a unified testbed facility.

Following, we present the architecture of Syndesi; a smart building IoT testbed deployed at the premises of University of Geneva that has been augmented with virtual resources. Using the physical component of the derived modelled testbed, a dataset consisting of 300'000 measurements has been built. Several numerical methods are used in order to extract the corresponding values that seed the simulation of the virtual resources. Their accuracy in capturing the actual conditions of the physical space is evaluated using K-fold cross validation. The results - including Mean Absolute Percentage Error (MAPE) levels below 7% - demonstrate that high accuracy can be achieved.

2 Related Work

Experiment driven research communities in the fields of computing and networking heavily rely on experimenting facilities. Researchers have been trying to alleviate the limitations that characterise testbeds mainly by federating individual facilities into meta-testbeds. This way, individual research teams can join forces and provide researchers with the ability to run broader and more diverse experiments by accessing several - potentially heterogeneous - testbeds.

Such federations are feasible with tools such as those introduced in [1]. The OneLab experimental facility, presented in [2], is a leading prototype for a flexible federation of testbeds that is open to the current Internet. GENI, the Global Environment for Networking Innovation [3], is a distributed virtual laboratory for transformative, at scale experiments in network science, services, and security. The Fed4FIRE federation framework [4] is gradually enabling experiments that combine facilities from the different FIRE research communities while the GEANT World Testbed Facility [5] focuses on regular computer networks. In [6], the IoT Lab platform is presented where several IoT testbeds across Europe are federated along with crowdsourced resources (e.g. smartphones) that are provided by the general public. In [7], authors study the technological issues related to the provision of a web-based simulation environment for supporting interactivity between remote scheduling and control systems and a locally resident simulation system. Finally, in [8] authors present SWiMNet; a framework for parallel simulation of wireless and mobile Personal Communication Service (PCS) networks, which allows realistic and detailed modelling of mobility, call traffic, and PCS network deployment.

All the above efforts as well as additional ones in the literature, focus either on federating different and potentially heterogeneous physical testbeds or on providing simulation frameworks as testbeds. On the contrary, our work focuses on the fusion of physical testbeds and simulation frameworks and in particular on how to utilize virtual resources in order to augment physical testbeds.

3 The Syndesi Testbed

The Syndesi testbed is a system and a platform comprised of heterogenous devices, sensors and resources focusing on the Internet of Things. The first version [9] was mostly focused on providing personalized services for smart environments combining sensor networks, electrical appliances, actuators and gateways via various communication protocols and technologies such as Near-Field Communication (NFC), Bluetooth, ZigBee, 802.15.4, 6LoWPAN etc. That first version, also referred as Syndesi 1.0, has been extensively updated over the last years to the Syndesi 2.0.

3.1 Syndesi 2.0

The scope of the Syndesi 2.0 is not only to provide efficient and smart services to its users but also to serve as system-as-a-service. Through multiple entry points, a large number of heterogeneous devices are able to interconnect with the testbed and provide their resources. The architecture is designed with scalability and interoperability in mind, which allows even mobile resources and data to be aggregated in it. Since the update, smartphone users equipped with the Syndesi 2.0 Android application, can contribute with data collected from smartphone sensors directly into the server's database. In addition, the Syndesi testbed is following a RESTful architectural approach providing interoperability between its resources, devices, services and the internet. Benefiting from Syndesi's REST-enabled services, external requesting systems are able to access and manipulate textual representations of resources exposed to the Web, using a uniform set of stateless operations.

The overall architecture of the Syndesi 2.0 testbed is shown in Fig. 1. The core of the Syndesi 2.0 testbed is behind the gateway. The role of this gateway is two-fold; first it serves as a connection point for all the elements and components as the backbone wireless sensor network (WSN), the mobile crowdsensing smart-phone application, the web etc. Second, the gateway is implemented on a Linux based machine which acts also as the heart of the system in which all the information from the various components are collected and stored. Since September 2016, this server automatically queries the testbed sensors for measurements, storing the sensed data in an SQL database as well as keeping track of active/inactive sensors. To this day, the above database contains more than 300,000 measurements and it is being utilized for improving virtual sensor value assignment.

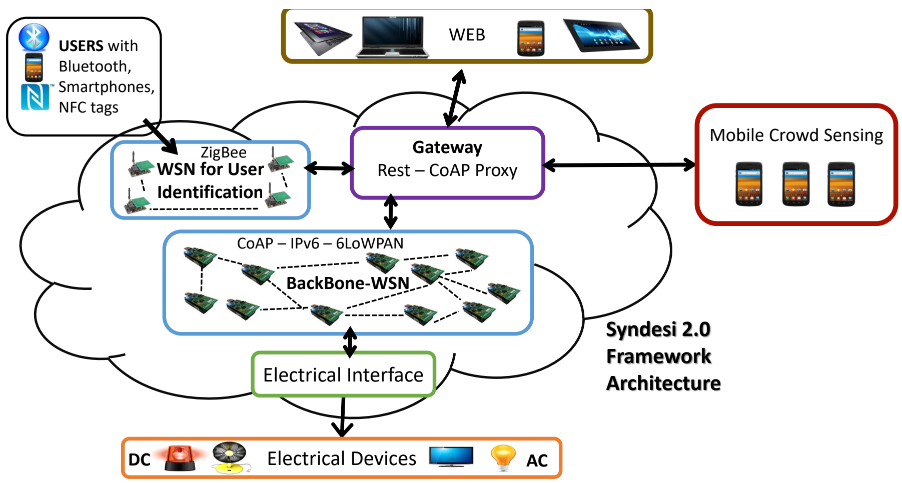

Fig. 1. Syndesi 2.0 framework architecture

3.2 Testbed Visualization

Traditionally, sensor testbeds are visually represented on top of floor plans, with sensor position information being 2D. This trend is also evident in sensor network emulator software, where visualization tools are often 2D only. We believe that 3D models are essential in order to store and represent models of testbeds, since algorithms that require spatial information (e.g. for distance calculation) will fail if the sensors are not co-planar, which is seldom the case in real world *wireless* sensor network deployments. We propose a simple model for storing the entire infrastructure, including buildings and interior objects spatial information as well as the sensors spatial information. For buildings representation, we store each block (either a wall, ceiling, floor or piece of furniture) by using 6 values: its x,y,z position and a value for scaling on the x,y,z axis. Using these 6 values, we can recreate rooms and interior objects which are in turn used to place sensors on. Regarding sensor information, only the spatial coordinates x,y,z are stored as well as the type of sensor.

The same model is used for storing information of physical as well as virtual sensors, making it easier for researchers to augment physical testbeds and use physical and virtual sensors interchangeably in algorithms. In Fig. 2 we see a screenshot of the interface that models Syndesi, which was developed with the help of the Unity engine [10].

4 Augmentation with Virtual Resources

When navigating in the 3D-visualization interface, a researcher can create virtual sensors to be added on the modelled testbed, which will have exactly the same properties and functionalities as the physical ones. The virtual sensors can be spawned anywhere a physical sensor could exist, i.e. in walls, windows etc, but not mid-air, and get their values from the physical ones via extrapolation techniques described in detail in the next section. Interaction with the virtual sensors such as drag and drop, sensor type configuration or sensor deletion are all provided by our interface. Once virtual sensors have been added in the 3D model of the physical testbed, the new augmented testbed can be saved as a separate instance, a function enabled by our back-end infrastructure, and then re-loaded in the future for further inspection/editing. This way, based on a single parent physical testbed, unlimited user-personalized modelled testbeds can be created to serve researcher demands.

4.1 Value Assignment

To calculate the values assigned to the virtual sensors, two approaches are followed. First, we look at the problem in a no-memory manner and we calculate the virtual values at any requested time using information only of the present moment, i.e. the values of all the physical sensors of the testbed. Then, with the end goal of improving the overall accuracy we make use of the collected dataset mentioned in section 3, in order to identify relational patterns between groups of physical sensors. These patterns are later used for associating newly spawned virtual sensors with specific subsets of the physical ones, resulting in more realistic value assignment.

Formally, we denote a physical sensor as:

$$s_i(x_i, y_i, z_i), \quad s_i \in S = \{s_1, s_2, ..., s_n\} \tag{1}$$

where x_i, y_i, z_i are the coordinates in space and S is the set of all n physical sensors that belong to the testbed. Respectively, a virtual sensor is denoted as:

$$v_i(x_i, y_i, z_i), \quad v_i \in V = \{v_1, v_2, ..., v_m\} \tag{2}$$

where the set V consists of the m virtual sensors created until that time in the modelled testbed. Sensor measurements for illuminance, temperature and

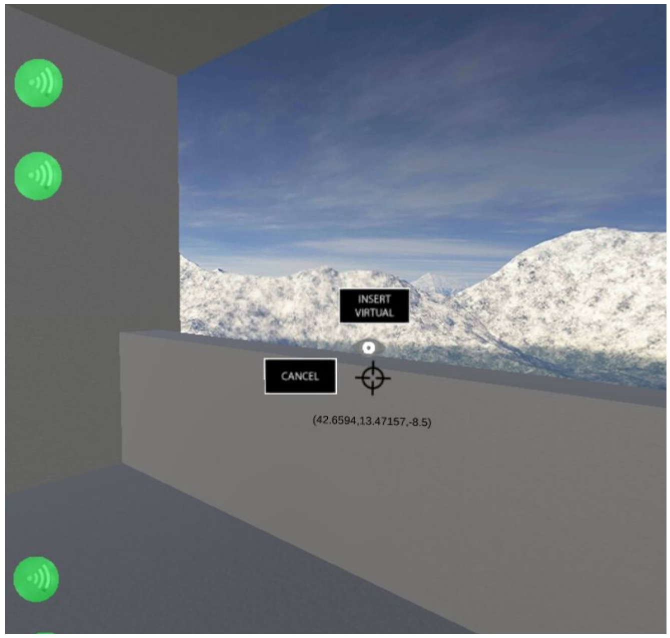

Fig. 2. Interactive interface of a modelled testbed

humidity are denoted respectively as $s_i(ill)$, $s_i(tmp)$, $s_i(hum)$ and likewise for virtual sensors.

When a virtual sensor v_k is spawned, e.g. with type illumance in an office A containing a physical sensors, its measurement value is calculated as a weighted average of the sensors located in the same office:

$$v_k(ill) = \frac{w_1 s_1(\dot{ill}) + w_2 s_2(\dot{ill}) + ... + w_n s_a(\dot{ill})}{\sum_{i=1}^{n} w_i} \tag{3}$$

We examined the use of the inverse and the inverse square of the euclidean distance between v_k and each of $s_i \in S$ as weights in the above equation, given that the spatial coordinates of virtual sensors are determined the moment it is created. This initial approach provided average results (MAPE levels over 20%) so we decided to use only a subset of the physical sensors $s \subset S_A$ to be taken into account in the calculation. In order to extract relevant subsets we effected the following brute force procedure using the past measurements stored in the server database:

1. Remove all measurements from a single sensor s_k.
2. Generate the powerset of $S - \{s_k\}$, i.e. all possible sensor subsets excluding the empty set.
3. Calculate the removed measurements via the weighted average using one subset at a time.

4. Compare calculated and actual measurements and keep the subset that produced the least absolute error.
5. Repeat for all $k \in \{1 \ldots n\}$

With the above method, we associate every physical sensor with a specific subset of the remaining sensors. That way, when a virtual sensor is spawned, depending on the proximity to the physical sensors, it receives its values based on the corresponding subset.

4.2 Evaluation of Accuracy

The dataset we conducted the evaluation comprises of a total of 295.536 measurements, 94.140 for illuminance, 103.062 for temperature and 98.328 for humidity collected from Syndesi during the period of 15-09-2016 to 31-12-2016. K-fold cross-validation was applied to evaluate the accuracy of the method, with the overall procedure depicted in Fig. 3. For each sensor type, the dataset was divided in 10, as close to equal, folds and each time the training process was applied in all of them but one which was kept for the testing phase. The process was repeated until all folds were selected for testing. It is worth mentioning here that given that a total of a sensors corresponds to $2^a - 1$ possible subsets the above brute force method does not scale well; in Syndesi testbed the experiments included 2 offices each containing 6 sensors, a setting which is computationally affordable, but for larger infrastructures different methods are envisioned

The results of the evaluation are shown in Table 1. First, we notice that there is minimal difference between inverse distance and inverse distance squared as a choice for weights in the calculation function, with the latter producing slightly better results. Overall, we observe very strong results regarding temperature and humidity measurements with Mean Absolute Percentage Error (MAPE) values around 1.5%. Such low values of MAPE ensure the credibility and scientific

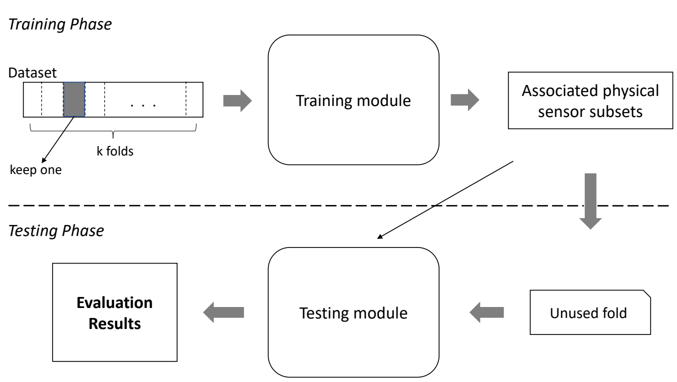

Fig. 3. Evaluation process

Table 1. Root Mean Square Error and Mean Absolute Percentage Error average scores after 10-fold cross validation.

Weighted average over inverse distance						
Sensor:	Illuminance(lux)		Temperature(^oC)		Humidity(%)	
Room	Average RMSE	Average MAPE	Average RMSE	Average MAPE	Average RMSE	Average MAPE
Office A	9.22 ± 0.55	3.02 ± 0.12%	0.33 ± 0.01	1.07 ± 0.02%	0.60 ± 0.02	1.50 ± 0.01%
Office B	44.4 ± 3.21	6.58 ± 0.20%	0.60 ± 0.01	1.91 ± 0.03%	1.17 ± 0.10	2.61 ± 0.05%
Weighted average over inverse distance squared						
Sensor:	Illuminance(lux)		Temperature(^oC)		Humidity(%)	
Room	Average RMSE	Average MAPE	Average RMSE	Average MAPE	Average RMSE	Average MAPE
Office A	9.39 ± 0.60	3.20 ± 0.14%	0.30 ± 0.01	0.98 ± 0.01%	0.58 ± 0.01	1.42 ± 0.01%
Office B	43.69 ± 3.41	6.57 ± 0.19%	0.61 ± 0.02	1.88 ± 0.04%	1.18 ± 0.10	2.62 ± 0.06%

correctness of experiments conducted in our modelled testbed. Illuminance is a tougher value to predict, as it can have highly uneven distribution in space, nevertheless the overall MAPE for all settings did not exceed 7% which is still a good result concerning the accuracy of virtual sensors. Finally, the noticeable difference between the two separate rooms regarding illuminance prediction (3% MAPE for office A and 7% for office B) can be explained due to different exposure in sunlight which result in discontinuities that are harder to predict.

5 Conclusion

In this paper we introduced the notion of modelled testbeds, which is based on the fusion of physical testbeds and simulation frameworks. The University of Geneva physical testbed Syndesi was modelled through an interactive GUI, where the addition of virtual resources is made possible for the users. To assign sensor measurement values to the virtual resources, several extrapolation methods were investigated and k-fold cross validation was performed to evaluate their accuracy. Results from experimentation in a dataset of 300'000 ambient temperature, illuminance and humidity measurements, collected from Syndesi, show that extrapolation from a subset of physical sensors based on proximity via weighted average over inverse distance squared provides optimal accuracy. As future work, we plan to extend the merging of physical and simulated environments to include the networking layer, thus enabling a broader range of experiments which can be conducted using modelled testbeds.

6 Acknowledgement

We would like to thank dr. Orestis Evangelatos and dr. Marios Karagiannis for their initial work and ideas on the deployment and expansion of Syndesi Testbed.

References

1. J. Augé, T. Parmentelat, N. Turro, S. Avakian, L. Baron, M. A. Larabi, M. Y. Rahman, T. Friedman, and S. Fdida, "Tools to foster a global federation of testbeds," *Computer Networks*, vol. 63, pp. 205–220, 2014.
2. L. Baron, R. Klacza, M. Y. Rahman, C. Scognamiglio, N. Kurose, T. Friedman, and S. Fdida, "Onelab tutorial: A single portal to heterogeneous testbeds," *EAI Endorsed Trans. Ubiquitous Environments*, vol. 2, no. 6, p. e4, 2015.
3. M. Berman, J. S. Chase, L. H. Landweber, A. Nakao, M. Ott, D. Raychaudhuri, R. Ricci, and I. Seskar, "GENI: A federated testbed for innovative network experiments," *Computer Networks*, vol. 61, pp. 5–23, 2014.
4. "Fed4fire project, https://www.fed4fire.eu/"
5. F. Farina, P. Szegedi, and J. Sobieski, "Géant world testbed facility: Federated and distributed testbeds as a service facility of géant," in *2014 26th International Teletraffic Congress (ITC), Karlskrona, Sweden, September 9-11, 2014*, pp. 1–6, 2014.
6. P. Alexandrou, C. M. Angelopoulos, O. Evangelatos, J. Fernandes, G. Filios, M. Karagiannis, N. Loumis, S. E. Nikoletseas, A. Rankov, T. P. Raptis, J. D. P. Rolim, and A. Souroulagkas, "A service based architecture for multidisciplinary iot experiments with crowdsourced resources," in *Ad-hoc, Mobile, and Wireless Networks - 15th International Conference, ADHOC-NOW 2016, Lille, France, July 4-6, 2016, Proceedings*, pp. 187–201, 2016.
7. S. Cavalieri, M. Macchi, and P. Valckenaers, "Benchmarking the performance of manufacturing control systems: design principles for a web-based simulated testbed," *J. Intelligent Manufacturing*, vol. 14, no. 1, pp. 43–58, 2003.
8. A. Boukerche, S. K. Das, and A. Fabbri, "Swimnet: A scalable parallel simulation testbed for wireless and mobile networks," *Wireless Networks*, vol. 7, no. 5, pp. 467–486, 2001.
9. O. Evangelatos, K. Samarasinghe, and J. Rolim, "Syndesi: A framework for creating personalized smart environments using wireless sensor networks," in *2013 IEEE International Conference on Distributed Computing in Sensor Systems*, pp. 325–330, May 2013.
10. Unity, a cross-platform game engine, Unity Technologies, https://unity3d.com/