Smart Environments

Dreossi, Tommaso
tom@commongarden.org

Hartnell, Jake
jake@commongarden.org

Hauk, James james@commongarden.org

DRAFT February 2018

Abstract

We propose a multi-faceted platform for smart environments which entails sensing, modeling, and controlling environmental variables. By using swarm technology, machine learning, and distributed ledgers, the Smart Environment platform we propose will allow growers and researchers (of all kinds, corporate, academic, citizen science etc.) to freely experiment, capture data, and *reliably* recreate their growing environments seamlessly.

1 Introduction

Smart environments can be formalized as Cyber-Physical Systems (CPSs), that are systems that integrate computations with physical processes [3, 14]. In our case, computations concern streams of data coming from sensors and actuators interacting with biological, hence physical, processes. In general, embedded computers monitor and control the physical processes, usually with feedback loops where physical processes affect computations and vice versa. The design of such systems, therefore, requires understanding the joint dynamics of computers, software, networks, as well as biological and physical processes [9].

We present a framework for creating such systems in the context of Agricultural and Biological research and production.

1.1 Problem

Two of the most important open problems in dealing with living organisms are *prevention* and *reproducibility*.

Prevention means being able to stop something from happening. Examples include the outbreak and spread of diseases (e.g., peronospora [20]) or the exhaustion of resources (e.g., water, fertilizer, etc.). Being able to forecast such harmful scenarios in advance gives the users the possibility to be prepared and avoid financial losses. Accurate forecasting methods [5] have been proven to be extremely useful for domains involving living organisms, e.g., plant epidemiology [7, 12], energy consumption [19] or productivity [17, 16].

Reproducibility is the ability to get the same results based on data previously gathered. Reproducibility is central in the scientific method [22] and many communities, from farmers [1] to computer science researches [21], are becoming sensitive to the ability of being able to precisely replicate experiences. The systematic reproduction of experiments also allows users to investigate the causes behind rare events and possibly replicate them. Examples are the exceptional maturation of a vegetable or the outbreak of diseases.

At the core of prevention and reproducibility are *data collection* and *analysis* [11, 8]. Data collection is the process of gathering and measuring information on targeted variables in an established systematic way. Data acquisition enables the analysis of an outcome and the answering of relevant questions, hence the prevention and reproduction of particular scenarios.

Growers and researchers often manually collect and organize data in notebooks and spreadsheets. The same data are then used to build ad hoc models for analysis and hypothesis that are biased by previous experiences and available hardware. Moreover, data is frequently collected from non-connected expensive hardware.

The lack of a rigorous, structured, and systematic data collection and analysis might penalize users and prevent them from optimizing their growing environments.

1.2 Solution

Smart environments are physical worlds that are richly and invisibly interwoven with sensors, computational elements, and actuators [24]. A smart environment usually encompasses three main elements:

1. **Data Collection -** Sensors are devices whose purpose is to detect changes in their environment and send information to other modules (e.g., computers). The data collected by sensors, that can be seen as the lifeblood of a smart environment;

2. **Data Analysis -** Computational elements are the brain of smart environments. Their task is to automatically analyze data generated by sensors and take decisions. The analysis and decisions are done by algorithms coming from Machine Learning (ML) [18, 4] — often erroneously called Artificial Intelligence (AI) [2] — and Control Theory [15, 6], respectively.

3. **Actuation -** The control decisions taken by the computational brain are sent to *actuators*, that are components responsible for moving or controlling a system that might alter the state of the smart environment (e.g., opening or closing a valve).

The client (grower, researcher) can remotely interact with the smart environment via an intuitive application that gives access to all data and devices. This enables remote and constant supervision regardless the client's location.

Grow-IoT comes with a suite of tools for monitoring, in real time all the data coming from the environment: that is sensed data, decisions statuses, and states of activators.[10]

2 Architecture

In this section we present how smart environments are formalized, organized, and implemented.

2.1 Discrete Events

We use *discrete time* models to describe the cyber-physical systems we are creating.[13] Discrete time is more suitable for our purposes since we deal with sensors and actuators that collect data and trigger actions at discrete time instances. For instance, a soil moisture sensor detects the soil humidity every ten minutes, or a water pump is either on or off at a specific time.

2.2 "Things"

We refer to the devices that compose our cyber-physical systems as "*things*". Each thing is an event emitter producing timestamped events [13]. Some examples of things in smart environments are, e.g.:

- Organisms (perhaps at least a virtual objects representing organisms), e.g., plants, bacteria, fishes;
- Sensors, e.g., moisture sensors, thermometers, barometers;
- Actuators, e.g., water pumps, light switches, fan activators;
- A connected device, to be actuated both from the Edge (directly) and the Cloud (remotely);
- Feedback algorithms developed and modified over time, designed to sustain life
- Groups or collections of things

Note that there might be many things concurrently emitting events.

Another interesting aspect is that there can be hierarchies of things and importantly decision making within cyber-physical systems.

A sensor or actuator by itself is not useful unless it is viewed in context. By collecting together event data from sensors, actuators, and living things themselves we can create meaningful linked data. Importantly, event types are namespaced. For example, a temperature event can have its value be Celsius, Kelvin, or Fahrenheit. Luckily, linked data handles this all wonderfully. Technologies such as JSON-LD [] can serve as namespaces for the events. Thus many types of things emitting many types of events can be integrated into our framework, in essence building complex interrelationships out simple components.

2.3 Reactive Modules

The events generated by things can be listened by *reactive module* that, driven by the observed data, make decisions and take actions. For instance, a reactive module might listen for temperature events coming from sensor arrays in a green house and control actuators such as heaters, fans, and vents.

Grow Files is a way of encoding information about the organism's basic requirements and its desired phenotype. It can be serialized in JSON, JSON-LD, or XML. When we start the Grow, a stream of new events is created from all the things emitting events (there may or may not be listeners).

A Grow File is composed by three main components: targets, cycles, and phases.

- *Targets* create listeners for events from sensors and emit alerts or correction events. For instance, if an ideal threshold is specified for a target, a controller, such as PID [23], emits correction events that drive the system towards the target. Corrections can be used to control, for instance, heaters, dosing pumps, etc.;

- *Cycles* are functions that are called at specific times in succession (for example, during the course of a day). Cycles are also a way of defining moving targets. For example, different target temperatures can fixed at daytime and nighttime;

- *Phases* are a way to create groups of cycles and/or targets. For instance, a plants life cycle might be broke up into the seeding, vegetative, flowering, and harvest phases. Each phase might have different environmental conditions with regards to lighting, pH, nutrients, temperature, etc. Phases may have a length attribute which specifies how long they last. Additionally, some cases may require human intervention to transition the grow system towards the next phase (such as transplanting seedlings, or replacing the water in the reservoir). In other words, phases may automatically or manually transition into the next phase.

Such a data model constitutes the reactive module, a stream of events from an environment (or collection of environments) is parsed and new events are triggered given what's defined in a Grow file. For example, a low temperature might cause a heater to turn on, as well as fans to circulate the air in the chamber.

2.4 Benefits

Smart environments can benefit growers in several ways.

Hardware *Minimal* and *modular* sensors and actuators rather than complex all-inclusive devices. This benefits the farmer is several ways:

1. *Cost reduction* - The grower buys only what is really needed.

2. *Extensibility* - The grower can buy any device whenever is needed and dynamically adapt the smart environment complexity to the scale of the farm.

3. *Longevity* - The equipment is upgradable at anytime.

Software Much of the data is automatically collected without human intervention. This relieves the grower from the tedious job of manually collecting numbers and enables the systematic, consistent, and interoperable organization of the gathered information.

Properly organized data are easier to visualize and represent, easier to be analyzed and parsed by machine learning systems, and easier to transfer and share across different farmers. Moreover, a standardized data collection can lead to a potential market place where farmers can share knowledge about their crops. This benefits both the single and the community.

Management The most important advantage provided by Common Garden is the way how smart environments are managed. The whole processed of data collection, analysis, decision making and actuation are automatized and data-driven.

The analysis of data is automatically carried out by *machine learning systems*, i.e., a suite of algorithms that can learn and make predictions on data without being explicitly programmed. The are several benefits in using machine learning instead of manual analysis. First, time and money consumption. The workload is delegated to the machine instead of being scrutinized by the farmer or by a hired data analyst. Second, mathematical optimization. Machine learning algorithms are the result of years of research and fine tuning. These algorithms are designed to find the optimal solutions to problems that can be solved by looking at data. Some optimization examples are the reduction of resources consumption, such as water or fertilizers, or the maximization of phenotypic outcomes. A machine learning algorithm can find the optimal solution where a human might not see it. Third, there is a subtle advantage in using machine learning: machines are unbiased. They exclusively rely on data. A farmer might be influenced by previous experience or other factors. Objective data-driven decisions taken by machine learning systems can be more precise and effective than human subjective choices.

The automatized supervision provided by machine learning implies less human intervention which benefits the farmer who can hire less employees, scale better its farm, and rely on constant supervision 24h/day 365days/year.

Despite the advantages provided by automation, the farmer should not lose control on its crops. For this reason, Common Garden develops a monitoring software through which the farmer can remotely inspect the status of the farm. The monitoring application can be used to check the data, the decisions that the machine learning systems are about to take, or schedule and run new processes. The user interface of this application is purposely designed to be simple and easy-to-use for non-technical users.

References

[1] Making agriculture research data more discoverable, reusable & reproducible. http://www.godan.info/blog-posts/making-agriculture-research-data-more-discoverable-reusable-reproducible. Accessed: 2018-01-23.

[2] What is the difference between artificial intelligence and machine learning? https://www.forbes.com/sites/bernardmarr/2016/12/06/what-is-the-difference-between-artificial-intelligence-and-machine-learning/383326b32742 . Accessed: 2018-01-23.

[3] Rajeev Alur. *Principles of cyber-physical systems*. MIT Press, 2015.

[4] Christophe Andrieu, Nando De Freitas, Arnaud Doucet, and Michael I Jordan. An introduction to mcmc for machine learning. *Machine learning*, 50(1-2):5–43, 2003.

[5] George EP Box, J Stuart Hunter, and William Gordon Hunter. *Statistics for experimenters: design, innovation, and discovery*, volume 2. Wiley-Interscience New York, 2005.

[6] Stephen Boyd, Laurent El Ghaoui, Eric Feron, and Venkataramanan Balakrishnan. *Linear matrix inequalities in system and control theory*. SIAM, 1994.

[7] C Lee Campbell, Laurence V Madden, et al. *Introduction to plant disease epidemiology*. John Wiley & Sons., 1990.

[8] Min Chen, Shiwen Mao, and Yunhao Liu. Big data: A survey. *Mobile Networks and Applications*, 19(2):171–209, 2014.

[9] Patricia Derler, Edward A Lee, and Alberto Sangiovanni Vincentelli. Modeling cyber–physical systems. *Proceedings of the IEEE*, 100(1):13–28, 2012.

[10] Common Garden. Grow-iot. https://github.com/CommonGarden/Grow-IoT, 2018.

[11] Tony Hey, Stewart Tansley, Kristin M Tolle, et al. *The fourth paradigm: data-intensive scientific discovery*, volume 1. Microsoft research Redmond, WA, 2009.

[12] L Huber and TJ Gillespie. Modeling leaf wetness in relation to plant disease epidemiology. *Annual review of phytopathology*, 30(1):553–577, 1992.

[13] Edward A. Lee. Modeling concurrent real-time processes using discrete events. *Annals of Software Engineering*, 7(1):25–45, Oct 1999.

[14] Edward Ashford Lee and Sanjit A Seshia. *Introduction to embedded systems: A cyber-physical systems approach*. MIT Press, 2016.

[15] Ernest Bruce Lee and Lawrence Markus. Foundations of optimal control theory. Technical report, MINNESOTA UNIV MINNEAPOLIS CENTER FOR CONTROL SCIENCES, 1967.

[16] Douglas C Montgomery, Lynwood A Johnson, and John S Gardiner. *Forecasting and time series analysis*. McGraw-Hill Companies, 1990.

[17] Steven Nahmias and Ye Cheng. *Production and operations analysis*, volume 4. McGraw-Hill/Irwin New York, 2009.

[18] Nasser M Nasrabadi. Pattern recognition and machine learning. *Journal of electronic imaging*, 16(4):049901, 2007.

[19] Alberto Hernandez Neto and Flávio Augusto Sanzovo Fiorelli. Comparison between detailed model simulation and artificial neural network for forecasting building energy consumption. *Energy and buildings*, 40(12):2169–2176, 2008.

[20] J Palti and Y Cohen. Downy mildew of cucurbits (pseudoperonospora cubensis): the fungus and its hosts, distribution, epidemiology and control. *Phytoparasitica*, 8(2):109–147, 1980.

[21] Roger D Peng. Reproducible research in computational science. *Science*, 334(6060):1226–1227, 2011.

[22] Allen F Repko. *Interdisciplinary research: Process and theory*. Sage, 2008.

[23] Daniel E Rivera, Manfred Morari, and Sigurd Skogestad. Internal model control: Pid controller design. *Industrial & engineering chemistry process design and development*, 25(1):252–265, 1986.

[24] Mark Weiser, Rich Gold, and John Seely Brown. The origins of ubiquitous computing research at parc in the late 1980s. *IBM systems journal*, 38(4):693–696, 1999.