Managing COVID-19 testing requires using Digital Twins

How the deployment of digital twin modeling can improve COVID-19 testing.

Zach Sullivan - Apr 21, 2020
Medium - Toward Data Science

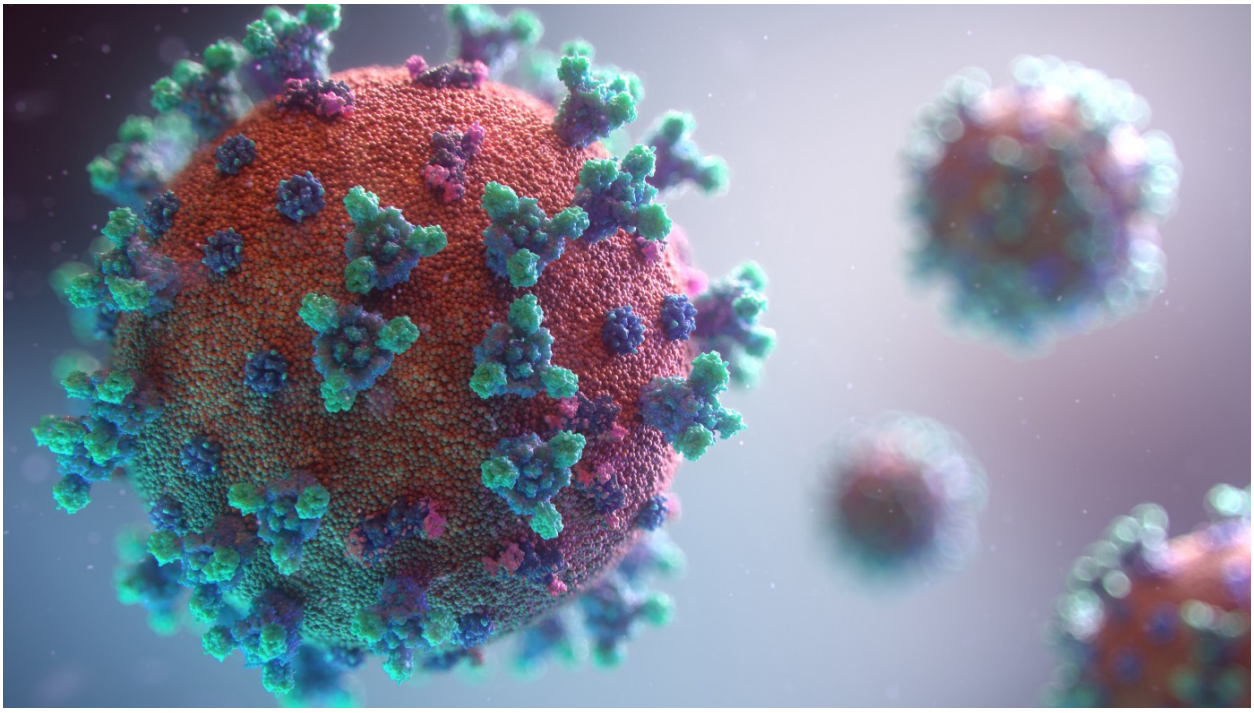


PHOTO BY FUSION MEDICAL ANIMATION ON UNSPLASH

Digital Twin technology has emerged from Internet-of-Things (IoT) and Cyber-Physical Systems (CPS) research [1][4] to provide a virtual representation of real-world physical assets or systems within an operational, high-fidelity digital model of the same physical entity and its surroundings.

In general, over the past 30–40 years products have grown more integrated with the Internet, where many products now have a dedicated media presence on the web. With the introduction of digital technology into traditional product cycles, digital modeling and simulation during both development and operations has proven to be a costly and slow process to mature. **Digital Twins** allow an organization to make better decisions about the design, management and performance of their human resources, procedures and assets, helping them to be more robust and flexible to change. Consequently, the value of **Digital Twins** across our society is enormous.

What is a Digital Twin

A **Digital Twin** allows system developers to better understand and optimize their physical-world programming through effective modeling of, for example, spatial orientation and movement, accurate operational performance characteristics, and material scheduling and sequence plan adaption, often using Machine Learning (ML) techniques. These digital representations may help anticipate and respond to failures through diagnostic (existing condition) / prognostic (future condition) analysis. Case in point. Consider a recent flight over the holidays from San Francisco (SFO) to Boston Logan (BOS) via a transfer at Washington Dulles (IAD) — from the online booking and billing, to the automated kiosk dispensing my boarding pass after providing the corresponding billing confirmation number, to the message board in Dulles alerting passengers on our late arriving flight that passengers with cross-connected flights would need to access the airline's kiosks in order to select a preferred flight connection option.

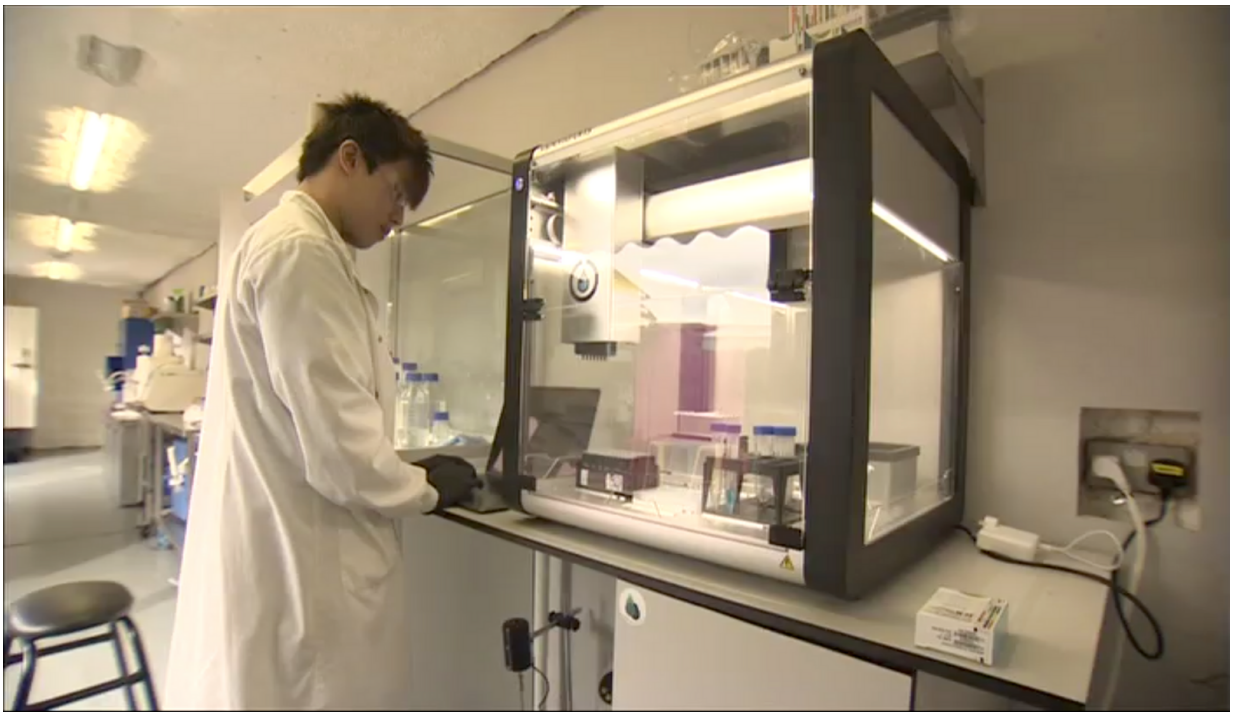
PHOTO BY DANIEL LIM ON UNSPLASH

All of this occurred, seamlessly, because the airline I flew that day had architected my whole travel experience through their system using a model of me — my airline **Digital Twin**. My twin tracked the time I started my trip (triggered when my first boarding pass was issued in San Francisco), to when I cleared security. The **Digital Twin** took into account when I actually boarded the aircraft and when it took off — even down to adjusting the amount of fuel loaded that day given the number of passengers, crew and cargo actually boarded. Once I was in the air, it followed my progress across the country. Indeed, as we waited for our delayed landing stacked up over Northern Virginia, my airline's **Digital Twin** also knew instantly that my progress was running late. Had I been a preferred customer of the airline (I just don't travel that much), it would have alerted me to the consequences of my lateness in-flight via email

suggesting that I needed to reschedule the onward travel to my destination (ultimately offering to do that automatically should I respond electronically with my preferences). Other than the security personnel, the flight crew and a few requisite people at the airline's arrival and destination gates, my passage was smooth and without any unnecessary transactions along the way. Historically, this has not always been the case.

Why is a Digital Twin important

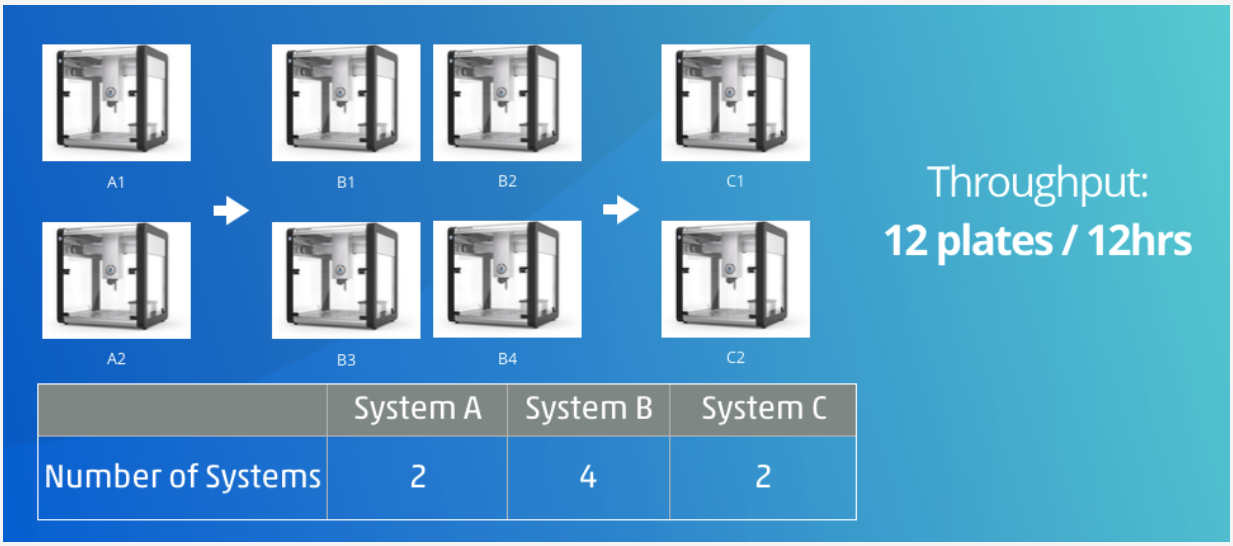
I suspect that many of the people reading this article today would not be particularly surprised to learn of this state of airline affairs, although few, I believe, would be aware of just how much of a 'tip of the iceberg' this airline case is when compared to broader application of **Digital Twin** technology across the whole of computing infrastructure globally. Again, for many airlines **Digital Twins** are not solely aligned to just their human customers but also includes **Digital Twin** models of their aircraft as well. Aircraft maintenance facilities regularly interact digitally with these physical assets both on the ground and in the air, often helping the aircrew to tune both the airframe and engine performance as an integrated system including fault diagnosis and support, occasionally even linking in the original equipment manufacturers as well.



OpenTrons liquid-handling robot *OT-2* (similar to many 3D printers) performing DNA test preparation at a ***OpenCell*** medical lab in the UK
source: OpenCell, UK

Operators of other large transportation fleets also regularly monitor individual vehicles as to their fuel loads, operational performance and location using standardized **Digital Twin** protocols [5]. In turn, many large manufacturing facilities have also learned to sensor-ize their larger, more expensive equipment so that its **Digital Twin** model can watch out for signs of early wear and tear that could result in costly downtime. Some of these **Digital Twin** models use sophisticated

signal analysis techniques to learn what are normal operating conditions and what is abnormal (e.g. bearing vibration, or material feed rates), from which the model can generate an equipment prognosis in order to schedule supportive maintenance. Further, several hospitals in the U.K. and Canada have been designed and built over the past decade, in response to earlier SARS and MERS epidemics, specifically to monitor — continuously from admission to release — their patient's condition (e.g. heart rate, blood oxygenation, test results, location and overall status) [7]. Here, these **Digital Twin** models provide critical situational awareness to aid in key roles such as synchronization of staff and resources when scheduling medical procedures, surgeries or laboratory work, while also distributing automatic alerts to unexpected situations in realtime.



Upper: PCR workflow graphic of ***OpenTrons*** COVID-19 testing process
Lower: ***Opentrons*** COVID-19 system configuration of 8 robots handling a 96 sample plate through the four step workflow
source: OpenTrons

How does a Digital Twin work

As has been highlighted repeatedly with the novel Coronavirus pandemic, laboratory procedures and testing are proving to be a bottleneck for many jurisdictions, especially when those tests use complicated DNA profiling and PCR amplification techniques that often involve a labor intensive sequence of steps that need to be performed with precision [8]. Many national healthcare systems worldwide are coming under increased strain to process DNA tests as quickly as possible (versus the emerging self-contained anti-body tests that confirm an existing infection after the fact). Within this context, a **Digital Twin** models the performance of one or many laboratory procedures operating in either stand-alone, serial or parallel configurations, offering rapid insights into potential operational problems alongside corresponding optimization and / or mitigation solutions in a reproducible manner. As described by Liu et al. 2018 [2] "The

digital twin is actually a working model of the physical asset or system, which can adapt to operational changes based on collected sensor data and information, and then forecast the future of the corresponding physical counterpart."

Enter **Opentrons**, who, from their location in Brooklyn, New York, have developed an open source liquid-handling robot called OT-2, suitable for a number of different tasks within medical laboratories worldwide. Thus, developing a **Digital Twin** that compliments the OT-2 robot would further aid the **COVID-19** testing process, allowing developers on their local computers to experiment with and evaluate new OT-2 testing procedures or even remotely monitor an OT-2 in real-time as it updates its Digital Twin model when the state of physical laboratory devices changes.

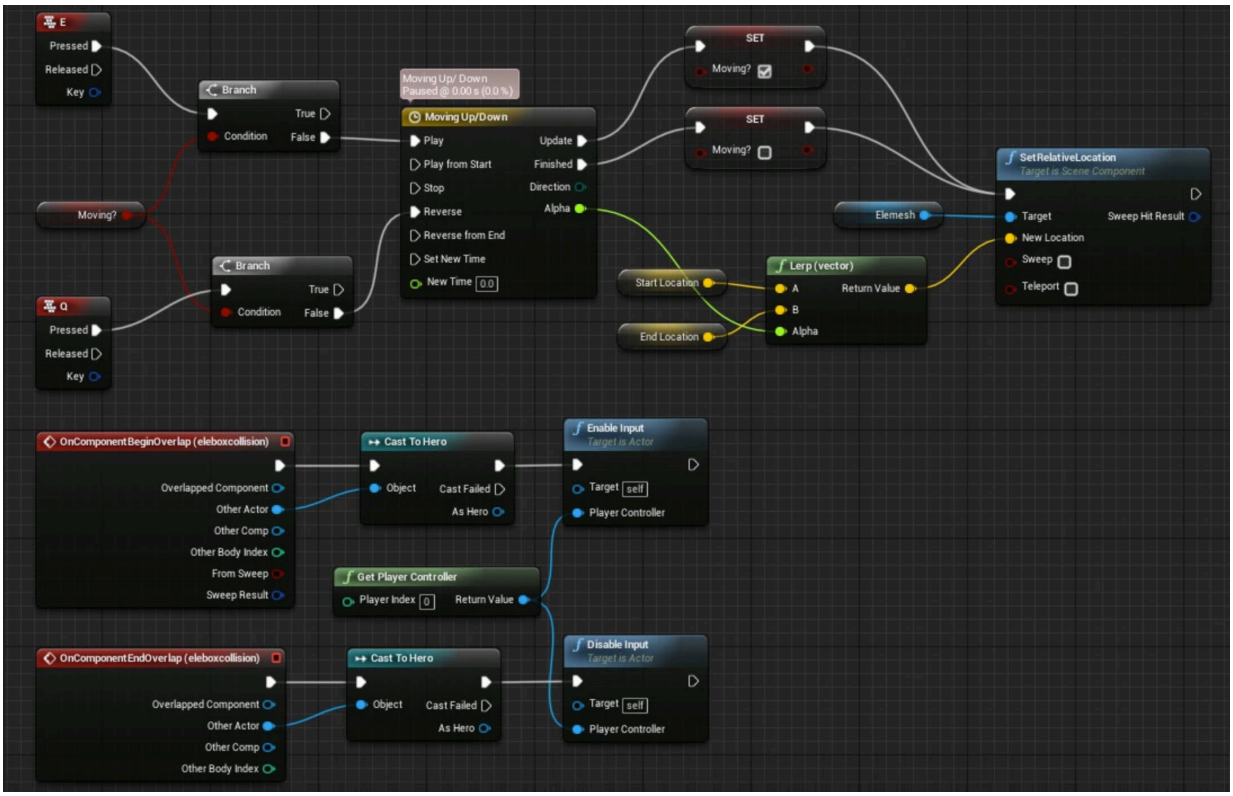
The intention of many medical facilities is to ramp up testing capabilities, usually from a range of approximately 500 tests per day to more than 10,000. Automating this testing is the best course of action to take as it offers the quickest way to correctly identify people who are infected with **COVID-19** and quarantine them for treatment without allowing the contagion to spread.

Epic Game's Unreal Engine 4 (UE4) user interface
source: unrealengine.com

Opentrons has been working with partners in the US, UK and Spain, among others, to use their OT-2 robot and machine operating software in a **COVID-19** test configuration of a four-step PCR assay process. Generally, a single test configuration comprises, at a minimum, a single bank of 4 robots aligned serially to process a workflow of up to 1150+ samples in a 24-hour period. All of this suggests that scaling these robotic test configurations towards processing well in excess of 100,000 test samples per day would require:

- multiple banks of robotic test configurations being managed to run semi-synchronously (i.e., automatic hand-off of a sample plate from one robot to the next);

- identifying and understanding (i.e., measure / record) **a)** time-motion aspects of individual test procedures and calibrated tool handling, **b)** sense / manage air handling conditions within each machine (e.g. air temperature, humidity, air pressurized HEPA[1] filtration), **c)** activating contamination mitigation and sanitization procedures;
- designing test procedures that optimize execution time of individual test sequencing while maximizing a robotic configuration's end-to-end throughput; and
- coordinating the available human resources at each configuration to effectively keep all these machines operational and stocked with consumable (e.g. pipette tips, reagents, etc.)



An example of UE4's ***Blueprints*** visual scripting environment with *Nodes containing* Functions and/or Variables interconnected by *Wires* carrying Signals and/or Events.
sources: unrealengine.com + fiverr.com / mimolol

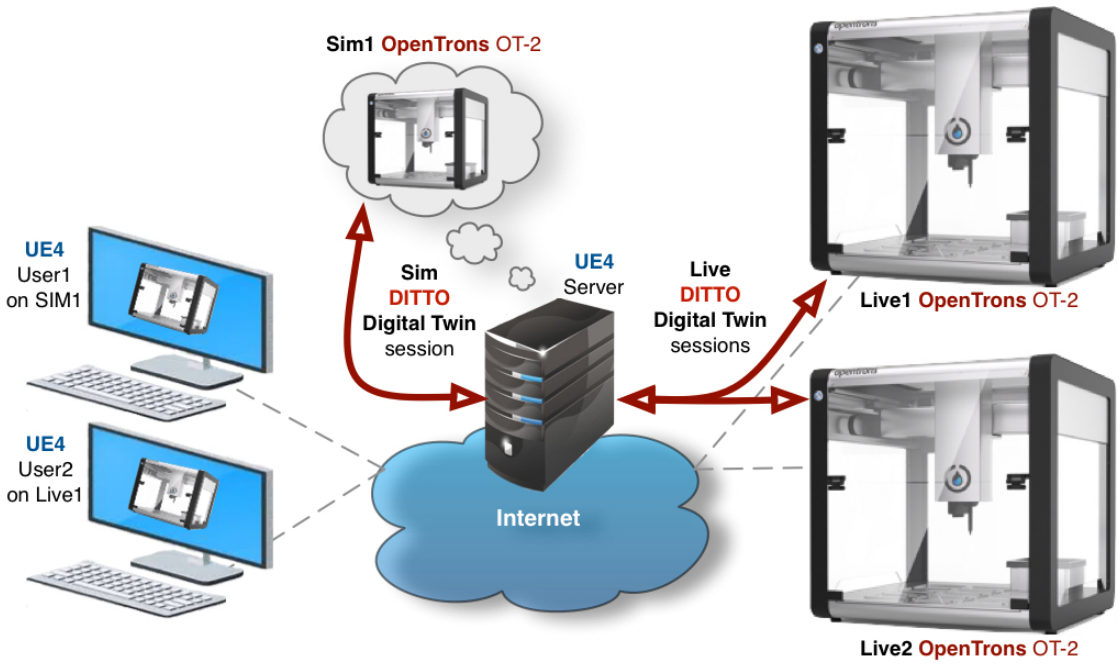
Building an OT-2 Digital Twin prototype

The emphasis in developing a **Digital Twin** model, mirroring the OT-2 robot, is that it helps to further enhance the speed and efficiency of clinical **COVID-19** tests and quality assurance procedures. The initial goal would be to produce a **Digital Twin** simulation that compliments the OT-2 robot by allowing developers on their local computers to experiment with and evaluate OT-2 handling [3]. Thanks to **Opentrons** open source policy, this functional simulation would leverage an accurate OT-2 robotic model (in

[1] **HEPA** = **H**igh **E**fficiency **P**articulate **A**ir filtration standard (EU & NA) which can remove 99.9% of air particles larger than 0.3 µm in diameter

terms of both motor behaviours and physical dimensions) running on the widely available game development platform — Unreal Engine 4 [6].

UE4 offers several key advantages over similar platforms, including a large and diverse user and developer community, readily available Open Source code with consistent licensing terms, a comprehensive Object Oriented design, an implementation with distributed and extensible networking support, and an advanced visual scripting (programming) environment called **Blueprints**.



OpenTrons OT-2 **Digital Twin** layout
source: ZS / OpenTrons

Blueprints is a complete scripting sub-system within **UE4** for creating and controlling game / play elements. Similar to many scripting languages, a **Blueprint** can define object-oriented (OO) classes or objects within the **UE4** engine. This system is quite powerful as it allows developers access to a wide range of software constructions and mechanisms generally only available to experienced **UE4** programmers. In addition, **Blueprint**-specific markup generated by **UE4** enables developers to create scripts that can be distributed and extended by others, from which an OT-2 community of developers can emerge. In a **Digital Twin** context, a **Blueprint** script controls one or more OT-2 robots. The primary benefit here is that it reduces the structural and syntactic details of a given programming language allowing non-professional programmers to think more in terms of modularity. Ultimately allowing contributors to focus instead on the flow of both control and data messages within their implementation.

Initially, the goal in pursuing this effort would be to simply build an accurate OT-2 simulation that would afford researchers and developers the opportunity to understand the issues in performing repetitive PCR tests. However, biomedical researchers often want to vary a test sequence for any number of practical reasons — which either requires engaging the able yet busy staff at **Opentrons** to modify tests or for individuals to develop their own test procedure code. From the latter, the obvious fact emerges that not all biomedical staff may have a

willingness to jump into a large body of software, and/or deal with often arcane programming languages. Equally, as testing infrastructure grows in response to demand, organizations need to effectively manage their collection of test machines, not only to minimize staff interactions and hence cost, contamination risk and potential errors, but also to achieve the highest available throughput as a return on their investment.

In considering these needs, the conclusion can be drawn that a “naive” simulation would likely prove not to be very useful. Consequently, this **Digital Twin** implementation would need to be redesigned in two specific ways:

1. Implement a communication protocol to connect with a real-world counterpart device, mirroring virtual devices residing in the digital world with physical entities in the real world. The communications protocol, DITTO [5] archives this, by working with either a “standalone” local simulation environment or connecting to a “live” OT-2 machine remotely — it allows developer’s test code to seamlessly interact with either; and
2. Including DITTO functionality, as a node, within the UE4 Blueprints visual scripting (programming) environment allows OT-2 test code to properly control the act of interrogating, retrieving or responding to data messages from one or more OT-2 machines concurrently, as well as easing biomedical staff programming tasks.

Working Forward

Digital Twins are affording researchers and developers a better way to optimize their physical-world projects through the virtual representation of these systems. In doing so, the **Digital Twin** model is capable of responding in real time to operational sensor data collected by its physical counterpart. Whether through the augmentation of modern air travel, the optimizations of fuel efficiency or the ability for hospitals to effectively monitor their patient's health conditions, **Digital Twins** continue to revolutionize our ability to make better decisions about the design, management and performance of our resources, procedures and assets.

With the observation that development of an accurate **Digital Twin** likely will take time, the potential to further enhance the speed and efficiency of clinical **COVID-19** tests and quality assurance procedures marks **Digital Twin** modeling as being of utmost importance. By leveraging the **Unreal Engine** development platform, combined with the communications protocol DITTO, a sophisticated **Digital Twin** mirroring the **Opentrons** open source liquid-handling robot can only help to further enhance the speed and efficiency of viral testing. In a time when many national healthcare systems are being put under increasing strain to process patient tests as quickly as possible, modeling and simulation offers an opportunity to rapidly gain insights into operational improvements alongside overcoming unforeseen problems.

Given the uncertainly of the current global situation, we will know if **Digital Twin** modeling has been successful when a community of developers forms in support of critical roles like this.

References

[1] Alam Kazi Masudul, El Saddik Abdulmotaleb, “C2PS: A Digital Twin Architecture Reference Model for the Cloud-based Cyber-Physical Systems” 2017, doi 10.1109/ACCESS.2017.2657006

[2] Liu, Zheng, et al. “The Role of Data Fusion in Predictive Maintenance Using a Digital Twin.” 2018, doi 10.1063/1.5031520

[3] Esther Landhuis, “Science and Culture: Putting a game face on biomedical research” — Proceeding of the National Academy of Science USA, 2016, doi 10.1073/pnas.1607585113

[4] ARUP, “Digital Twins: towards a meaningful framework” 2019, web: www.arup.com/digitaltwin

[5] Eclipse, DITTO Digital Twin protocol 2020, web: www.eclipse.org/ditto/intro-overview.html#what-is-it

[6] Epic Games, Unreal Engine 4 developer site, 2020 web: www.unrealengine.com/en-US/

[7] Health Canada, “Learning from SARS.” 2003 https://www.canada.ca/content/dam/phac-aspc/migration/phac-aspc/publicat/sars-sras/pdf/sars-e.pdf

[8] New York Times, “C.D.C. Labs Were Contaminated, Delaying Coronavirus Testing, Officials Say” 2020 https://www.nytimes.com/2020/04/18/health/cdc-coronavirus-lab-contamination-testing.html

Zachary Sullivan is a Harvard graduate student in software engineering, with a particular passion for developing “serious gaming” applications that address real world problems.