WiP Abstract: BraceForce: Software Engineering Support for Sensing in CPS Applications

Xi Zheng, Dewayne E. Perry, Christine Julien
The Center for Advanced Research in Software Engineering
The University of Texas at Austin
jameszhengxi@utexas.edu,perry@mail.utexas.edu,c.julien@utexas.edu

Developing and deploying CPS applications involves a large amount of low-level programming that requires interacting with different (often proprietary) data formats, languages, and operating systems. In practice, applications are built for specific sensor network platforms with little potential for portability to other platforms or integration with other sensors. Debugging CPS applications that inherently integrate with a physical environment requires not only the aforementioned integration of the application with sensing but also the use of a testing harness that, at debugging time, accesses sensed data about the physical environment for the purposes of validating the actions of the application. Integrating sensing for application and debugging support in a way that is easy, flexible, and portable is essential for supporting CPS application development.

We are building BraceForce [1], a middleware for CPS application development that simplifies the development, deployment, and debugging of CPS applications. In supporting application *development*, BraceForce allows separation of the application developer from the low-level interaction between the platform and sensors. In supporting application *deployment*, BraceForce allows the developer to connect the application to sensing assets in the deployment environment. In supporting application *debugging*, BraceForce can be used to monitor a test environment using capabilities that may not be available in the deployment environment. As is depicted in Fig. 1, BraceForce defines functional *tiers* that encapsulate related aspects but coordinate in such a way that the tiers' deployment to particular physical assets is flexible. Different tiers can be deployed on user-facing devices or on sensing devices with limited capabilities, directly addressing and leveraging the specific capabilities and intentions of each device. BraceForce provides auto-discovery of new sensing and computing assets, allowing easy integration of new capabilities into existing applications or automatic and seamless extension of a debugging environment.

BraceForce supports both the traditional *pull* and more flexible *push* based interaction with sensing devices. Out of concerns for energy-saving, BraceForce also supports the use of Model-Driven Data Acquisition (MDDA) to suppress sensor readings that are *predictable* according to some *a priori* or learned model. By default, BraceForce implements libraries that enable simple instantiation of a few MDDA models (e.g., temporal models based on sensing history), and it also provides flexible programming interface for the user to define customized models.

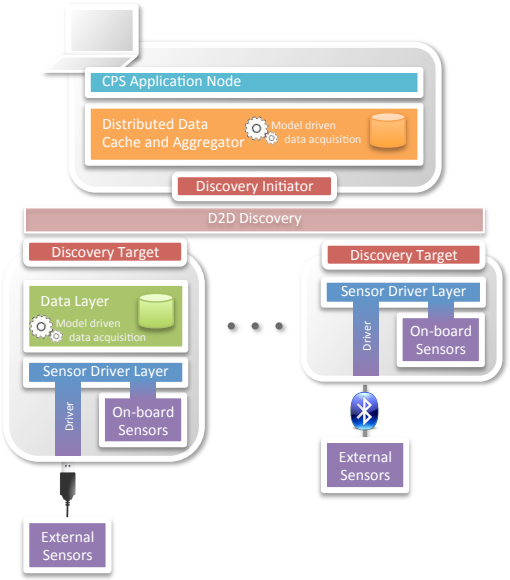

Figure 1: BraceForce: Mixed Environment

A preliminary user study has demonstrated that BraceForce can significantly simplify development of sensor-aggregation applications for novice developers. The work in progress includes demonstrating the BraceForce middleware in a real application (e.g. autonomous robots coordinated patrol) where the middleware can be ported to a different platform so that its performance and scalability impacts can be further studied. We are also investigating correlations between sensor data and the MDDA models used, and we are enhancing BraceForce's auto-discovery mechanisms across multiple-gateways.

Acknowledgement. This work was supported in part by the NSF under grant CNS-1239498.

1. REFERENCES

[1] X. Zheng, D.E. Perry, and C. Julien. BraceForce: A middleware enabling novice programmers to integrate sensing in cps applications. Technical Report TR-ARISE-2013-003, UT Austin, 2013.