103

104

105

106

Implementing Cross-Platform Protocol Execution with the Protocol Activity Modeling Language

Bryan Bartley^{1,*}, Jacob Beal¹, Alexis Casas², Jeremy Cahill³, Timothy Fallon⁴, Daniel Bryce⁵, Robert P. Goldman⁵, Luiza Hesketh⁶, Tim Dobbs⁷, Alejandro Vignoni⁸

¹Raytheon BBN (Cambridge, MA, USA), ²Imperial College London, ⁴Scripps Institution of Oceanography / University of California San Diego, ⁵SIFT, LLC (Minneapolis, MN, USA), ⁶Campinas State University, ⁷Learning Planet Institute (Paris, FR) ⁸Universitat Politécnica de València

bryan.a.bartley@raytheon.com

1 INTRODUCTION

1

2

3

4

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

Laboratory protocols are used for a wide range of purposes in research and development, at many different stages, including experiment design, execution, data analysis, interpretation, and communication and sharing with other groups (Fig 1). However, protocols are often difficult to communicate or reproduce, given the differences in context, skills, instruments, and other resources between different projects, investigators, and organizations. To this end, the Bioprotocols Working Group (https://github.com/Bioprotocols has developed a draft specification [1] for a unified protocol modeling language, called the Protocol Activity Modeling Language (PAML). The PAML data model has been designed to support the following needs:

- Execution by either humans or machines
- Maintaining execution records and associated metadata markup
- Mapping protocols from one laboratory environment to another
- Recording modifications of protocols and the relationship between different versions
- Verification and validation of protocol completeness and coherence
- Planning, scheduling, and allocation of laboratory resources

Here we describe recent progress implementing PAML and demonstrating that it can be translated to and executed across different laboratory platforms in order to address use cases presented by the stakeholder community.

2 AUTHORING AND EXECUTING PAML PROTOCOLS

PAML protocols may be authored using either the pypaml
programming API or the web-based PAML Editor (https:
//pamled.sift.net). The pypaml library implements the PAML
standard in Python and provides modeling, execution, and
exporting functionality. The PAML Editor is a React and
Django application built upon pypaml. It provides a browserbased visual scripting interface, using activity blocks whose

input/output ports are connected by flows. The PAML Editor is intended for a broad user-base and does not require significant programming expertise.

We have also implemented an execution engine as part of the pypaml library that can directly execute a protocol through an instrument's API or via translation into an executable format, such as Autoprotocol. The PAML execution engine uses a token based execution semantics that implements the UML activity model based upon Petri-nets [2]. Execution involves identifying when each activity should be executed and processing the flow of input and output tokens between activities.

To export protocols to other formats, the execution engine uses custom listeners that translate protocols. Current listener prototypes focus on laboratory automation (Autoprotocol) and human-readable "paper protocols" (Markdown). For example, the Autoprotocol listener translates PAML protocols into a list of instructions in JSON format operating on reagents and containers, and the Markdown listener makes use of Markdown syntax to hyperlink definitions of reagents and containers. Currently, members of our working group are also implementing specialized listeners for Opentrons and Echo lab robots.

3 EXAMPLES OF CROSS-PLATFORM PROTOCOL EXECUTION

Much recent PAML development has been driven by the International Genetically Engineered Machines (iGEM) community. This year, as part of the iGEM interlaboratory study, teams will exchange DNA constructs and run measurement assays to assess the reproducibility of the protocol. For this purpose, we have encoded several PAML protocols for calibrated fluorescence measurement in 96-well microplate cultures. This year students will be executing "paper protocols" generated as Markdown from PAML source. This "crowdsourcing" will serve as practical validation of PAML's ability to capture and describe relevant details for experimental execution and reproducibility.

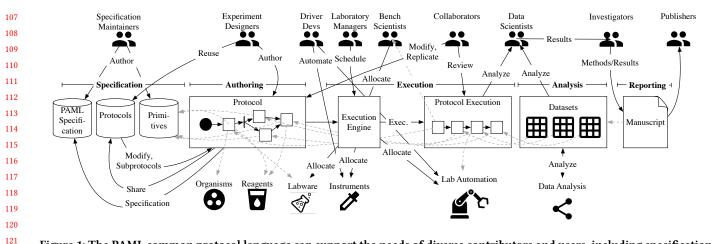
Looking forward to iGEM in 2023, our goal is to validate cross-platform execution of PAML by running these 

Figure 1: The PAML common protocol language can support the needs of diverse contributors and users, including specification maintainers, experiment designers, driver developers, laboratory managers, bench scientists, collaborators, data scientists, investigators, and publishers (top). Protocols can incorporate a variety elements, including activities, control flow, organisms, reagents, labware, scientific instruments, laboratory automation, and data analysis algorithms (bottom). It enables users to communicate about the operations involving these elements and partially or fully automate their design, execution, analysis, and interpretation (middle).

same protocols on Opentrons machines. Toward this end, the "Friendzymes" iGEM team, which represents an international, distributed, open science collaboration has been developing the Opentrons execution engine.

Another iGEM initiative related to protocol-sharing has grown out of a prior collaboration between the Imperial Col-lege London, Paris-Bettencourt, and Costa Rica teams. These teams have developed an automated Golden Gate assembly protocol with Beckman Coulter Echo[™] 525 / 550 acoustic liquid handlers. These teams are currently encoding their protocol in PAML and implementing a new cherry-picking listener to the execution engine.

Eventually, these common molecular biology protocols
will be disseminated as PAML on the iGEM Technology Resources web page for use among future iGEM teams and
future members of the remote automation collaborative network.

Our working group has also been collaborating with the consortium for Standardization in Lab Automation (SiLA). To demonstrate PAML capabilities for industrial control, we have developed a pH calibration protocol. This protocol il-lustrates the capability to adapt a base protocol with variable inputs (such as the volume of solution to produce) to multiple scales. It relies upon computational and physical primitives that calculate quantities, select appropriate labware and in-struments, mix materials, and alter execution based upon instrument telemetry.

These ongoing efforts to develop executable PAML protocols for a variety of platforms will provide a demonstration that the language can address many diverse use cases and challenges faced by existing interlaboratory collaborations.

4 FUTURE DIRECTIONS

We hope these initial demonstrations will convince others in the broader community of the value and utility of this emerging protocol standard and encourage others to contribute. To this end, our Bioprotocols Working Group is open to any organizations and individuals with an interest in standardized representation of biological protocols. We are currently drafting a formal organization, governance, and fundraising strategy for this community.

Acknowledgements

We would like to thank the Interlaboratory Working Group of the iGEM Engineering Committee for their help refining PAML protocols for use by the iGEM community. This work was supported by Air Force Research Laboratory (AFRL) and DARPA contract FA8750-17-C-0184. This document does not contain technology or technical data controlled under either U.S. International Traffic in Arms Regulation or U.S. Export Administration Regulations.

REFERENCES

- BARTLEY, BRYAN ET AL. Building an open representation for biological protocols. *bioRxiv* (2022).
- [2] PETRI, C. A. Communication with automata. PhD thesis, Universitat Hamburg, 1966.