

A Universal Language and Format to Describe Behavioral Tasks and Data

MOTIVATION

Technological advances have revolutionized how we measure and manipulate brain activity, while behavioral technologies have lagged behind. Recent developments introduced methods for quantifying movements and poses but we lack general approaches to describe and communicate behavioral tasks, which are necessary to infer internal states not visible from movements alone. Different laboratories use different systems, hardware, and software to probe behavior, making it difficult to communicate task design, share data, or reproduce experiments. Furthermore, neural data archives require matching behavioral data archives for interpreting neural activity. Here we developed a universal framework for designing, implementing, communicating, and archiving behavioral tasks.

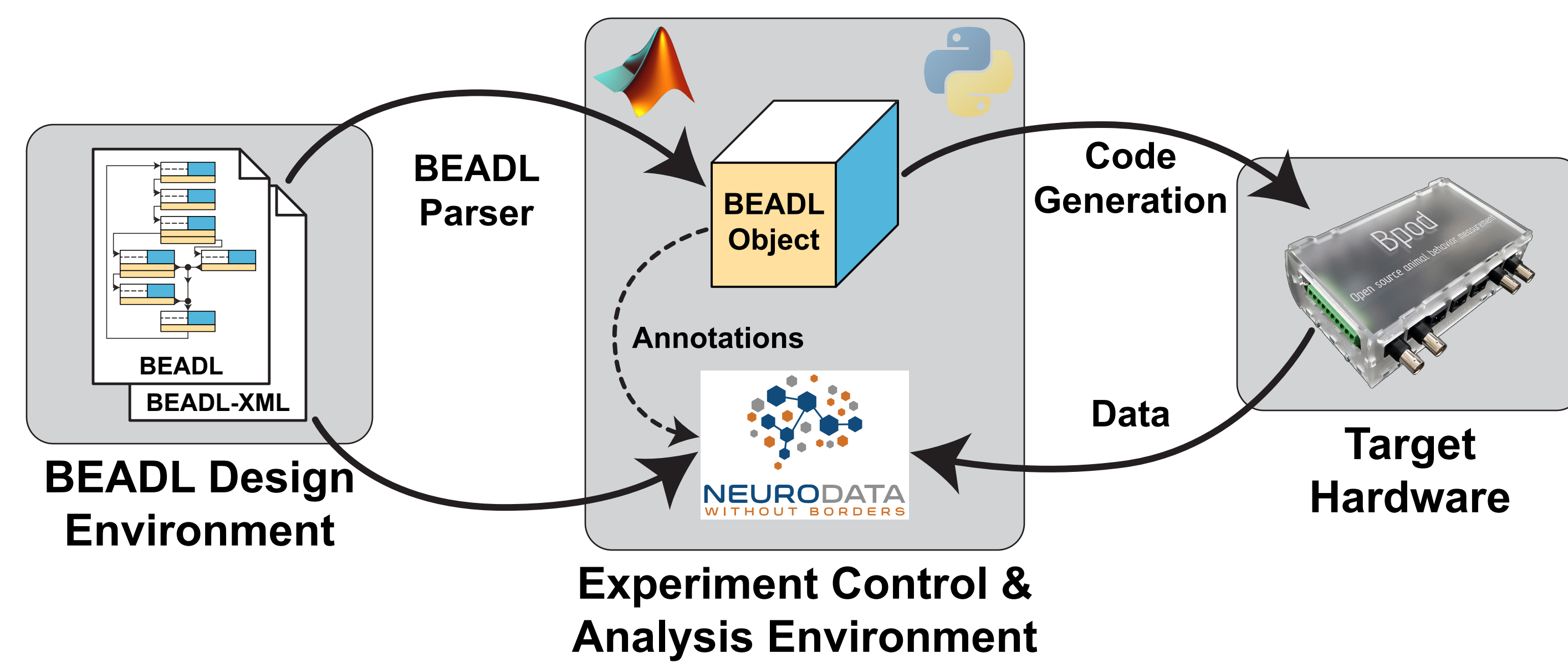
APPROACH

Our framework consists of two components, a description language and a data format along with associated software tools. BEADL, the BEhavioral tAsk Description Language, defines behavioral tasks as virtual finite state machines that can be described graphically as an easy-to-understand flow diagram. In each state, the sensed behavioral output of a subject is defined as events, causing transitions to other states. In addition, each state has a defined list of distinct actions, that the task controlling environment is performing (e.g., stimulus presentation). We use virtual inputs to generalize the descriptive power of this framework. BEADL's graphical representation can be exported as a corresponding XML-based definition. An NWB extension (Neurodata Without Borders) allows for storing of behavioral data capturing both the BEADL task description together with the behavioral output of a subject.

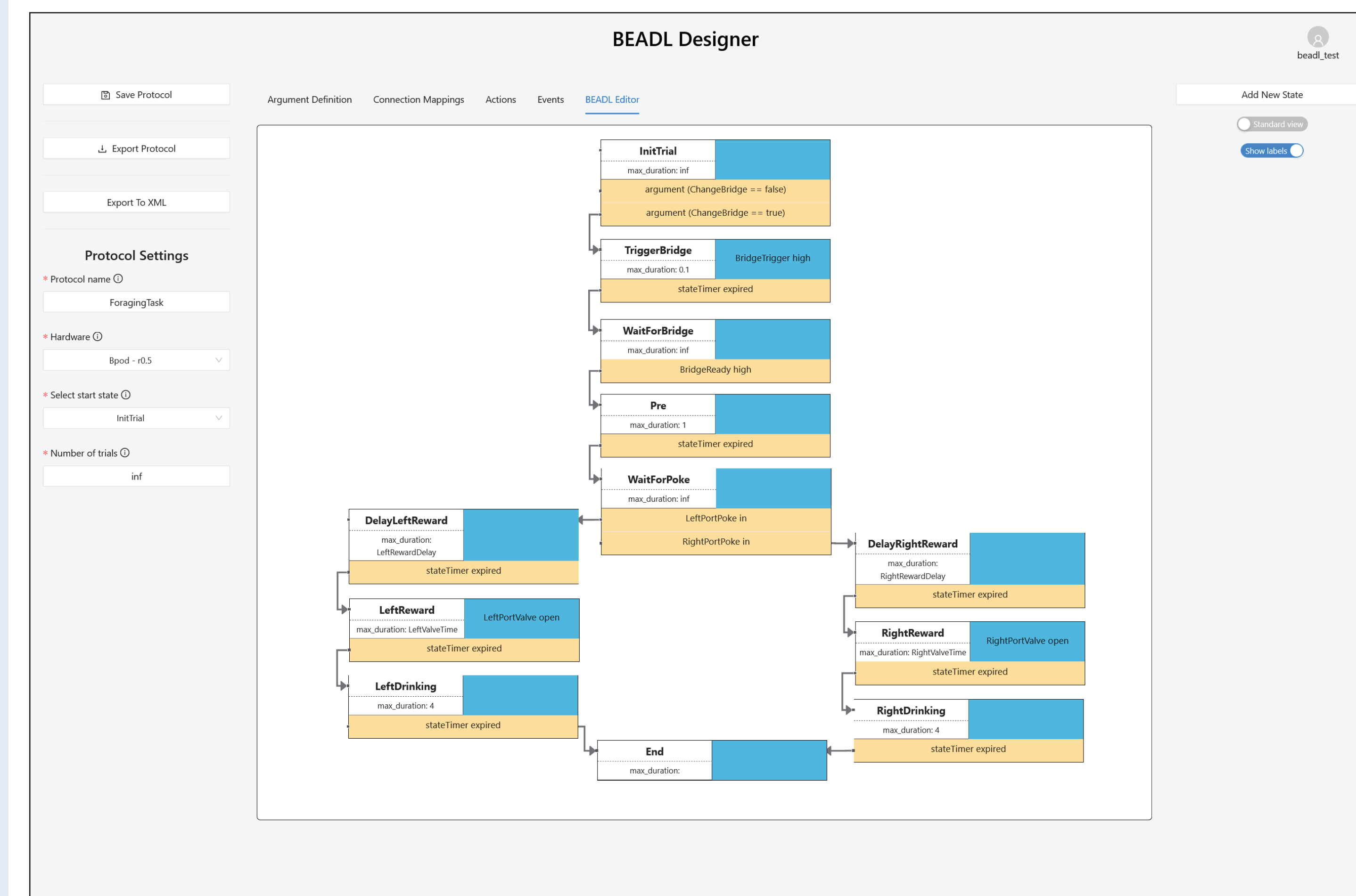
CONCLUSION

- BEADL's graphical representation of task descriptions simplifies designing and communicating trial-based behavioral tasks
- Inherent logic of behavioral tasks is exposed to ensure clarity, precision, details in task designs, communication, and reproducibility
- Web-based BEADL graphical task designer with integrated BEADL-XML generator
- NWB extension to further standardize data formats and to consistently linking neural and behavior data with the contingencies to the behavioral task
- This standardized framework allows more transparency and reproducibility in behavioral experiments

BEADL Design Workflow

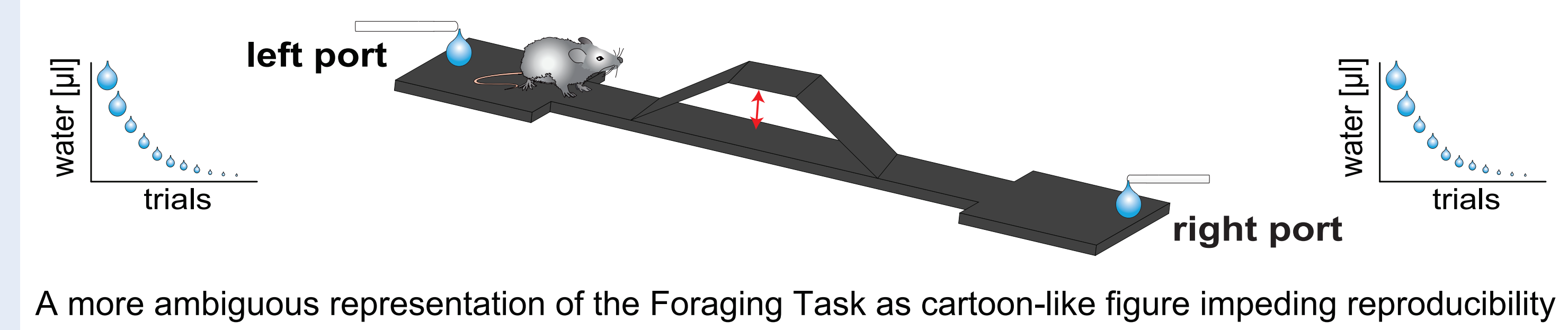


BEADL Designer



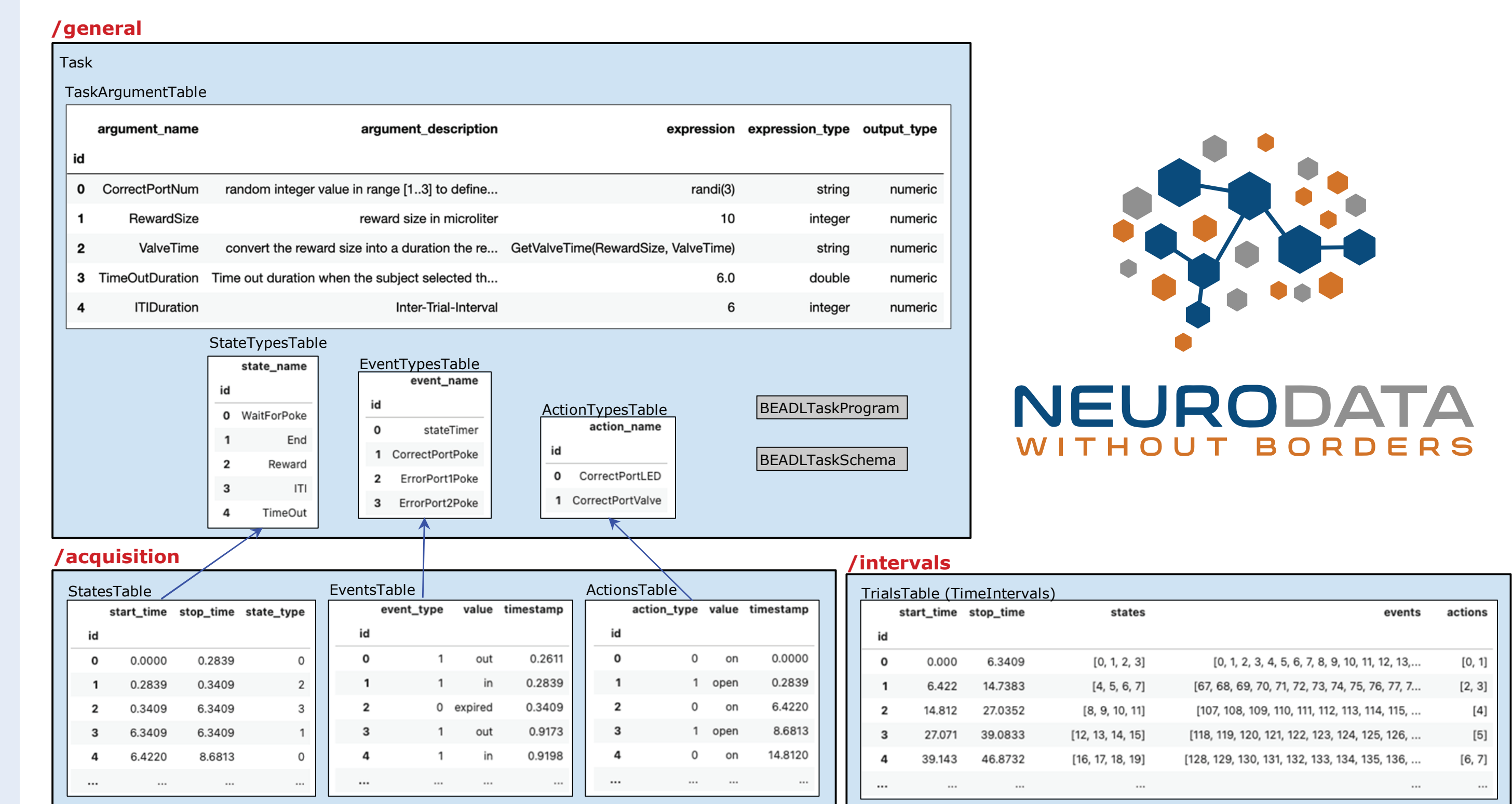
The Foraging Task as a BEADL diagram

Example of Abstract Task Representations



A more ambiguous representation of the Foraging Task as cartoon-like figure impeding reproducibility

Neurodata Without Borders (NWB) Extension for BEADL



- NWB extension stores BEADL task XML and schema alongside behavioral data associated with trials, states, and events in an NWB file
- Python API provides functions for importing Bpod data corresponding to a BEADL task description and querying the BEADL state/event definition from a recorded state/event

Resources

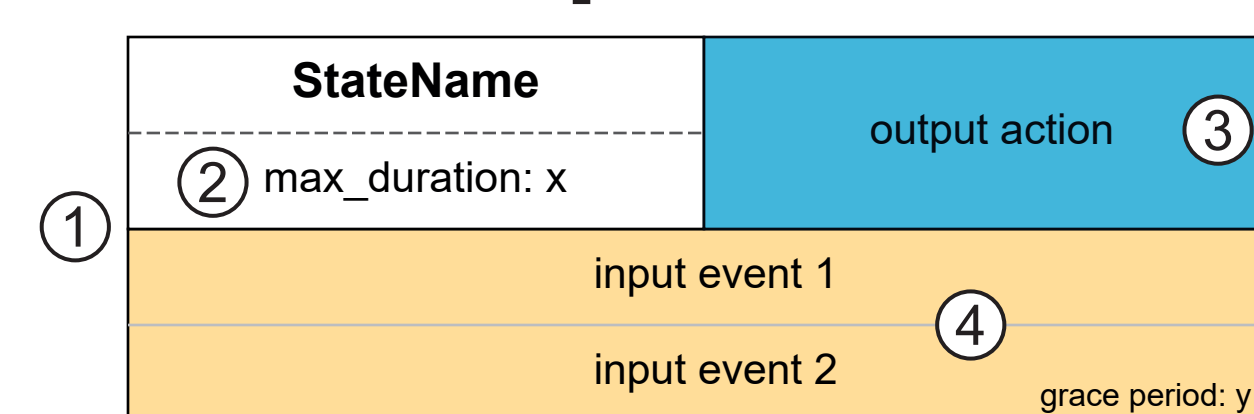
Please visit beadl.org for additional information, documentation, and upcoming features!



Funding



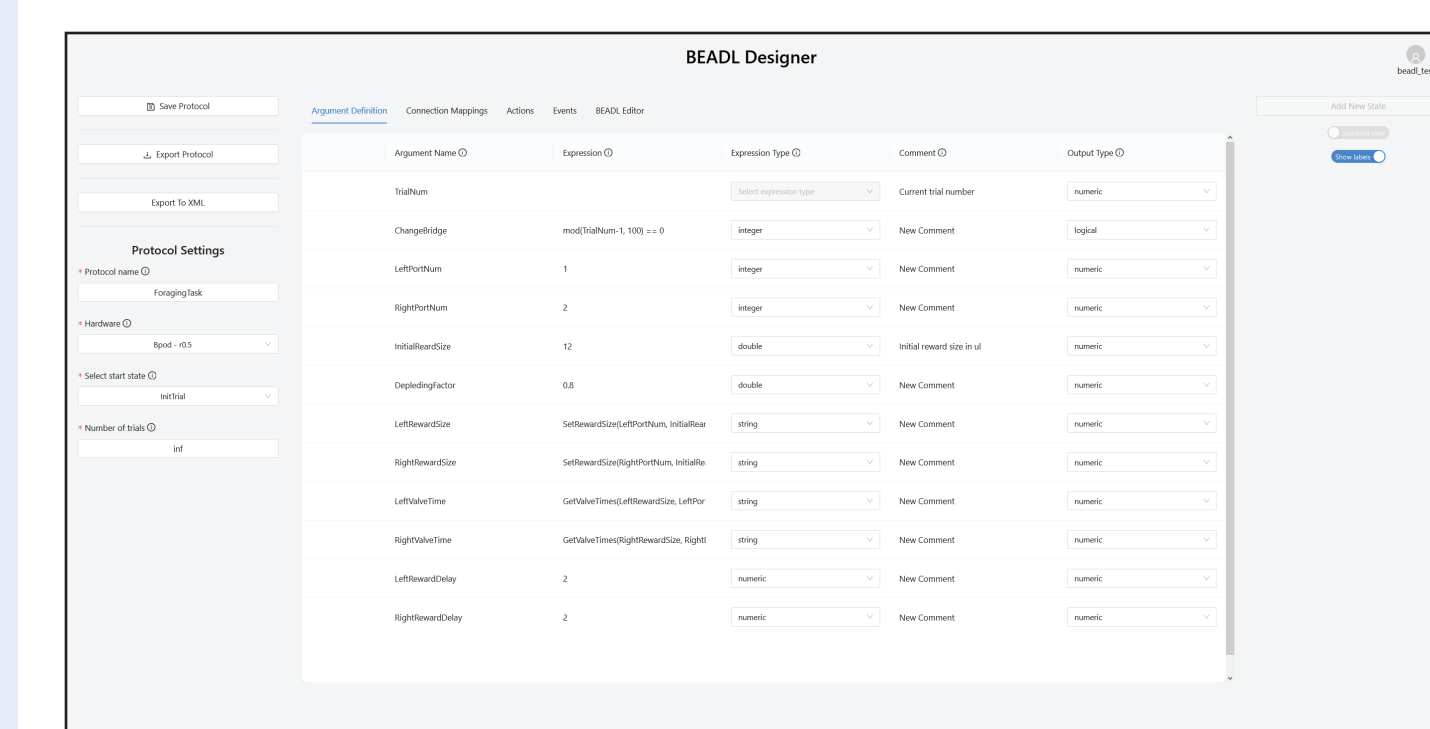
BEADL Graphical Elements



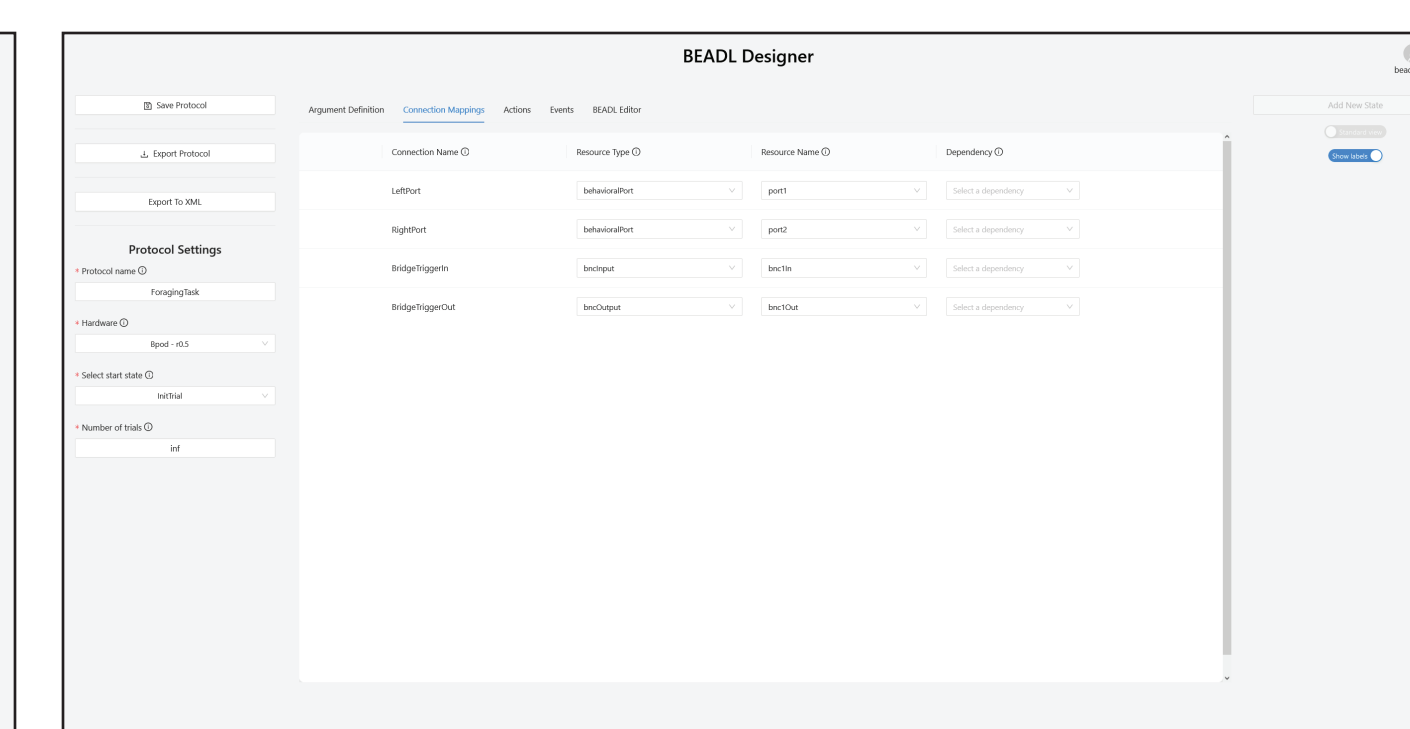
- ① The default BEADL state definition. Each state has a name, a max. duration, a list of output actions, and a list of input events.
- ② Each state can have a maximum duration specified in seconds. If it expires, a specific **stateExpired** event will be triggered. If the duration is set to infinite (**inf**), this feature is disabled. Trial-dependent parameters can also be used to define the duration of a state.
- ③ In each state, an arbitrary number of output actions can be defined that the experiment controlling hardware can perform (e.g. stimulus presentation). Besides those "basic" action, more hardware-oriented advanced actions can also be defined in form of callback functions.
- ④ For each individual state, specific input events can be defined that cause transitions to other states when they occur. Those events can either be inferred by the subject's behavior or by pre-defined conditions (e.g. the maximum state duration).

BEADL Graphical Design Application

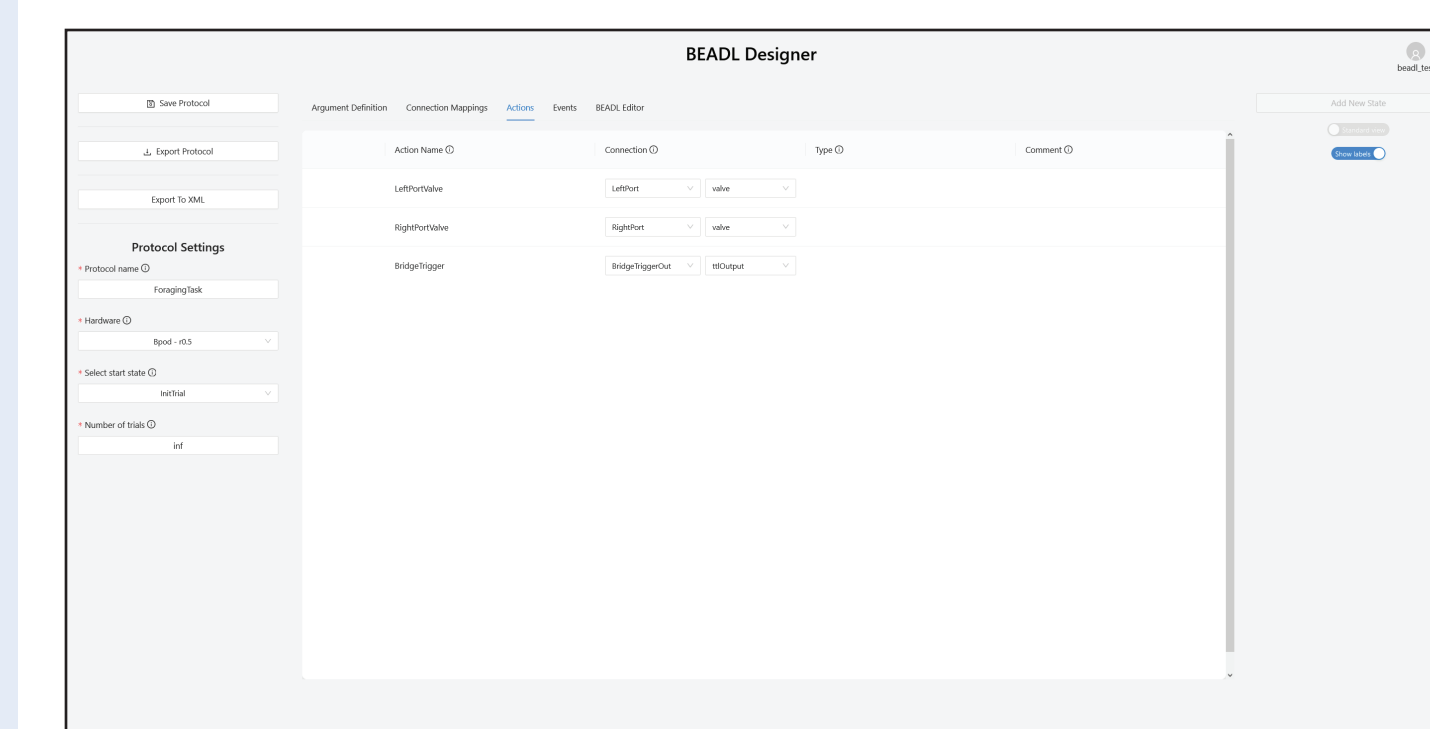
- Web application running on Amazon Web Services (AWS)
- Easy-to-use, drag-and-drop graphical editor to define behavioral task as BEADL diagrams
- Individual user workspaces to store different behavioral tasks
- Pre-defined BEADL Use Cases as templates available
- Definition of trial-dependent parameters
- Abstracting control hardware elements, actions, and events through descriptors
- Optional constraints with dependencies to trial-dependent parameters
- Export BEADL-XML description to be used for task executing (currently for Bpod/MATLAB)



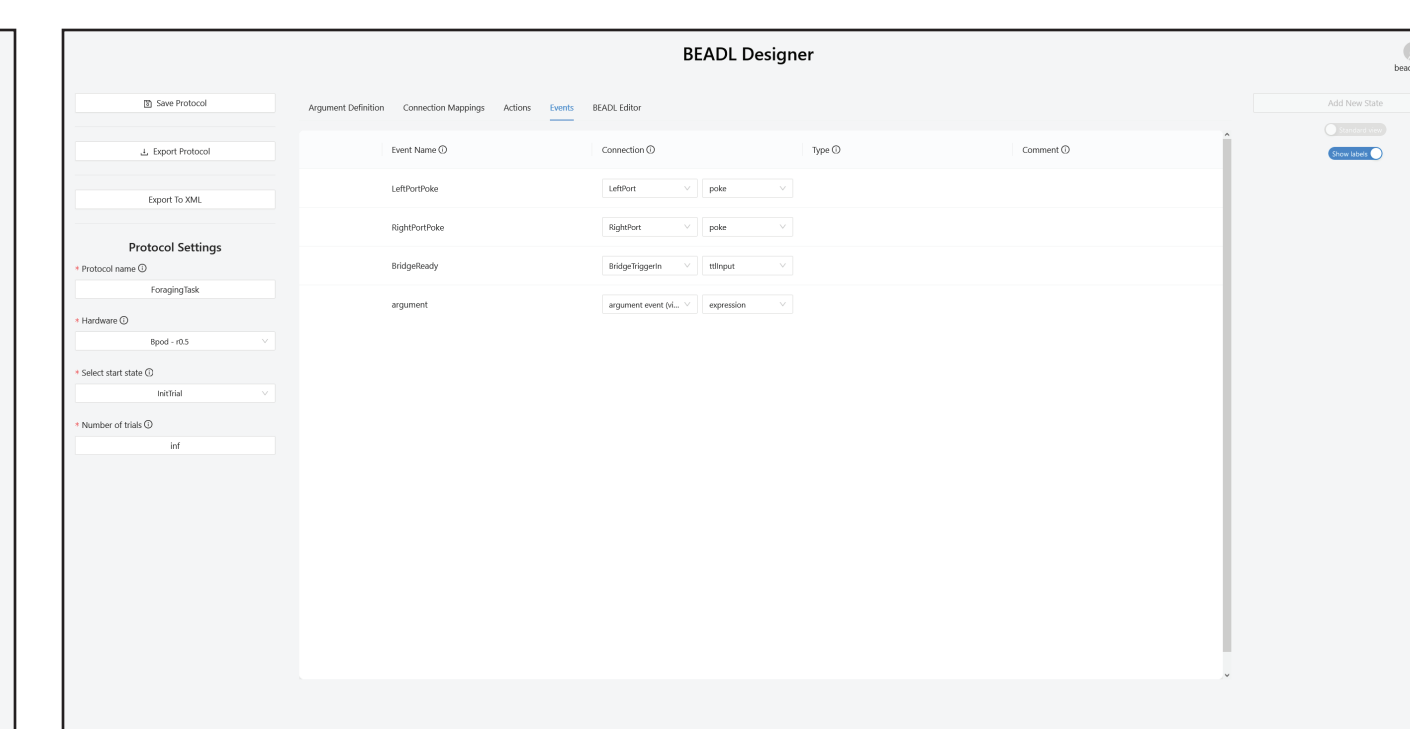
Definition of trial-dependent parameters



Mapping of hardware resources onto descriptors



Definition of actions being performed by the task-controlling system during task execution



Definition of input events (actions performed by the subject) to trigger state transitions