Imitation Platform User Guide

This is an all-purpose, step-by-step guide to using the Imitation Platform. Though functionality may be extended in the future, this guide will primarily reference usage with regards to the SMILE environment.

# Application Overview

The general flow of the application can be broken into a few steps as indicated in Figure 1. The application has five steps, one of which is optional.

# 

Figure 1. Imitation Platform Flow Diagram

# Name Task

A simple field to name the task to be imitated. Once you fill the field with a valid name, click Next.

# Generate XML (Optional Step)

This optional step has been implemented in order to generate the Initial XML State required for the last step. To generate a valid XML state, the user will provide a text file dictating the parameters for the XML generator (See Figure 2). Please see the documentation on this functionality entitled **Create User Input XML Explanation.docx** for more information. The generated XML will be placed in your current directory. Once your XML has been generated, click Next.

You may continue past this step without any action if you already have a valid Initial XML State by just clicking Next.

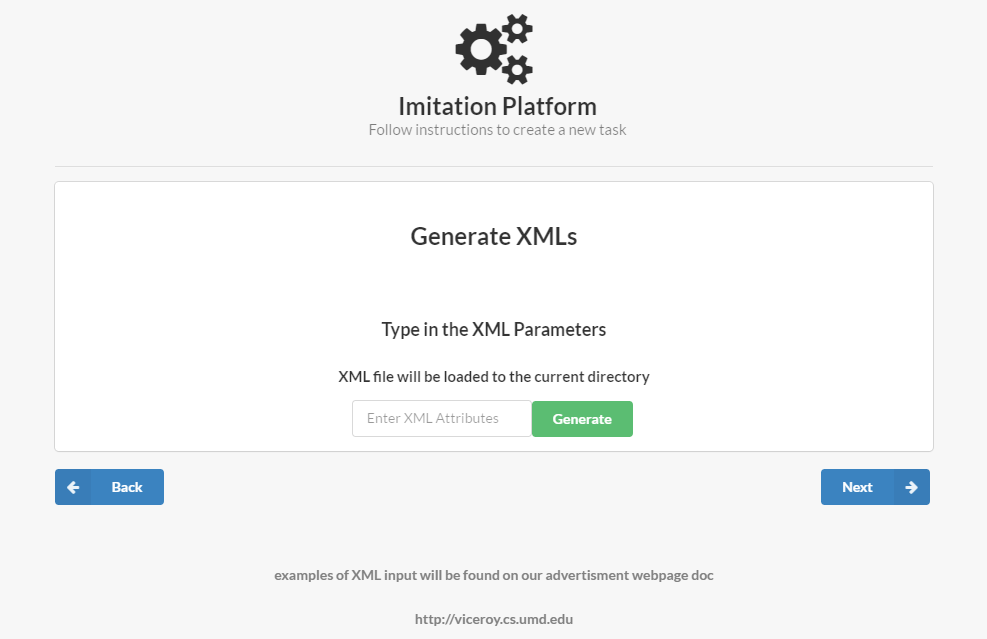


Figure 2. Optional XML Generator Page

# Insert Recording

This step requires that the user enters a SMILE recording of the action to be imitated (See Figure 3). Once the file has been selected, click Next.

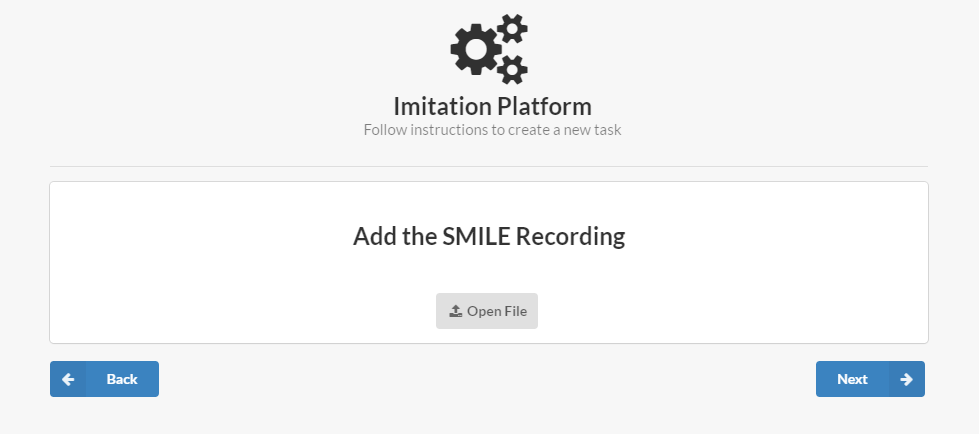


Figure 3. Provide the SMILE recording

# Knowledge Base

The causal knowledge for the imitation algorithm is provided in this step. The user has two options: build a custom knowledge base via the provided GUI or insert their own knowledge base script.

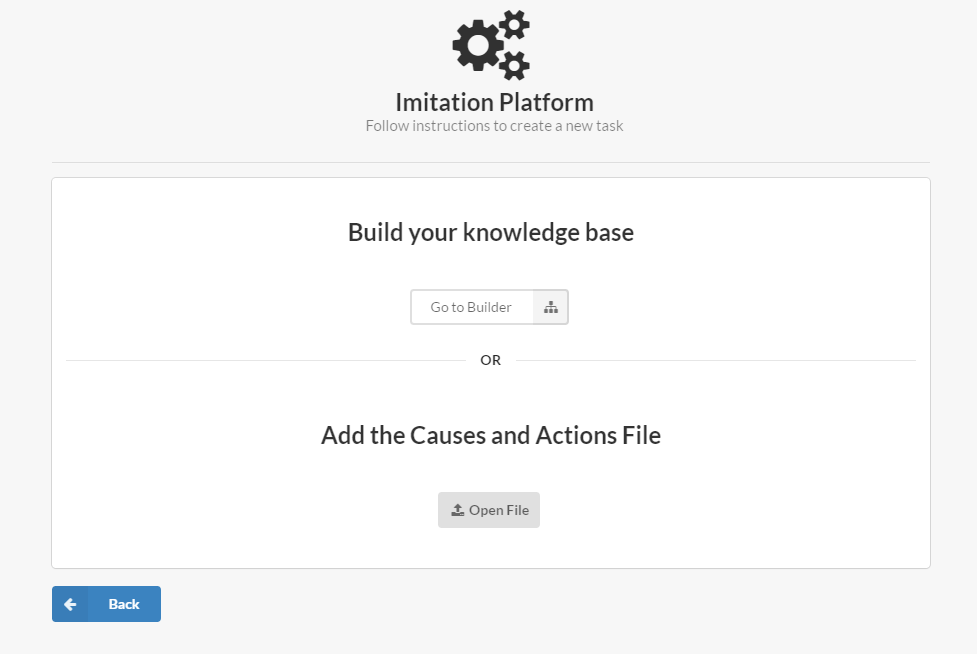


Figure 4. Knowledge base selection prompt

#### Build a Custom Knowledge Base

Begin building your knowledge base by selecting the Go to Builder option and proceed through the following steps:

1. Define Causes and Actions
   1. Create a cause by clicking the green plus button and giving it a name.
   2. Give your cause its component actions by clicking the green plus on the right and giving each action a name.

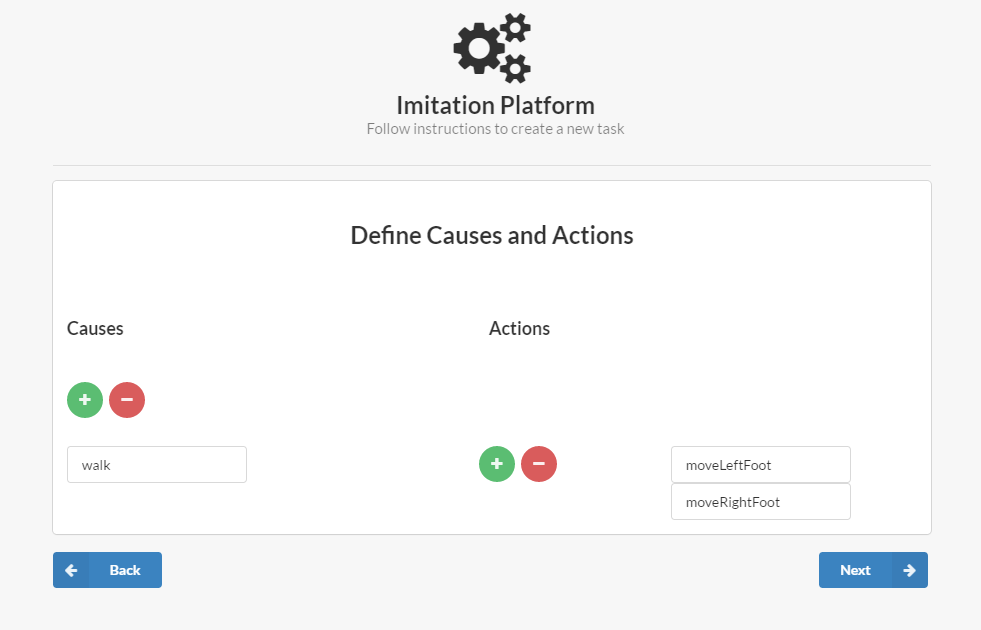


Figure 5. Define Causes and Actions

1. Define Causal Relationships
   1. Define the relationship between each cause and its actions that you defined in the previous step.
   2. Each relationship may be defined as Direct or Conditional.
      1. Direct Relationships:  
         These relationships hold unconditionally.
      2. Conditional Relationships:  
         These relationships only hold under certain conditions. You must define these conditions in the appropriate field. These conditionals take the same for as conditionals as described in the Causal Compiler Syntax Guide. An example conditional might look like

TYPE(obj) = block

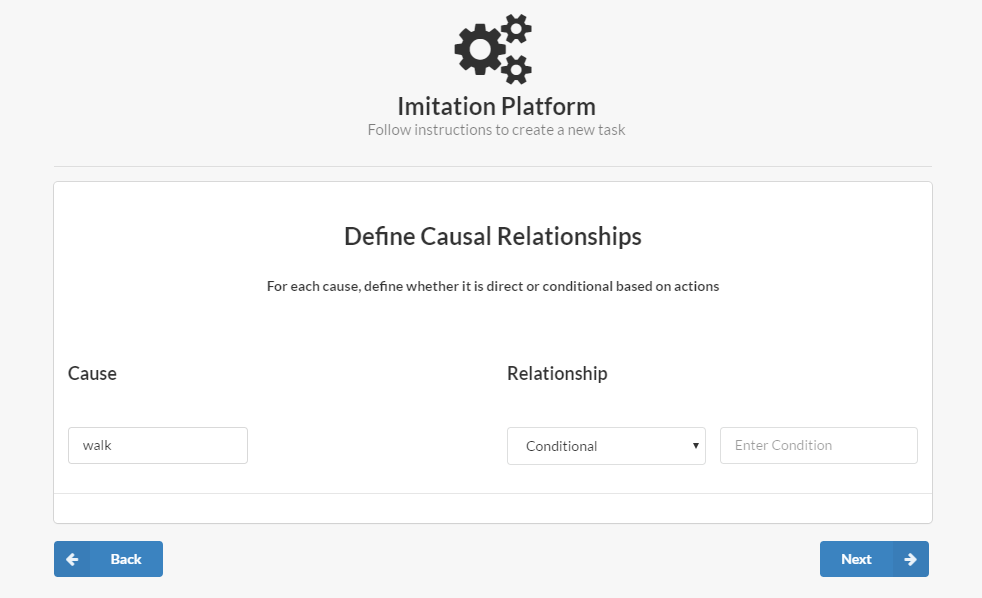


Figure 6. Define Causal Relationships

1. Define Action Parameters
   1. Define the parameters of each action that you previously defined.

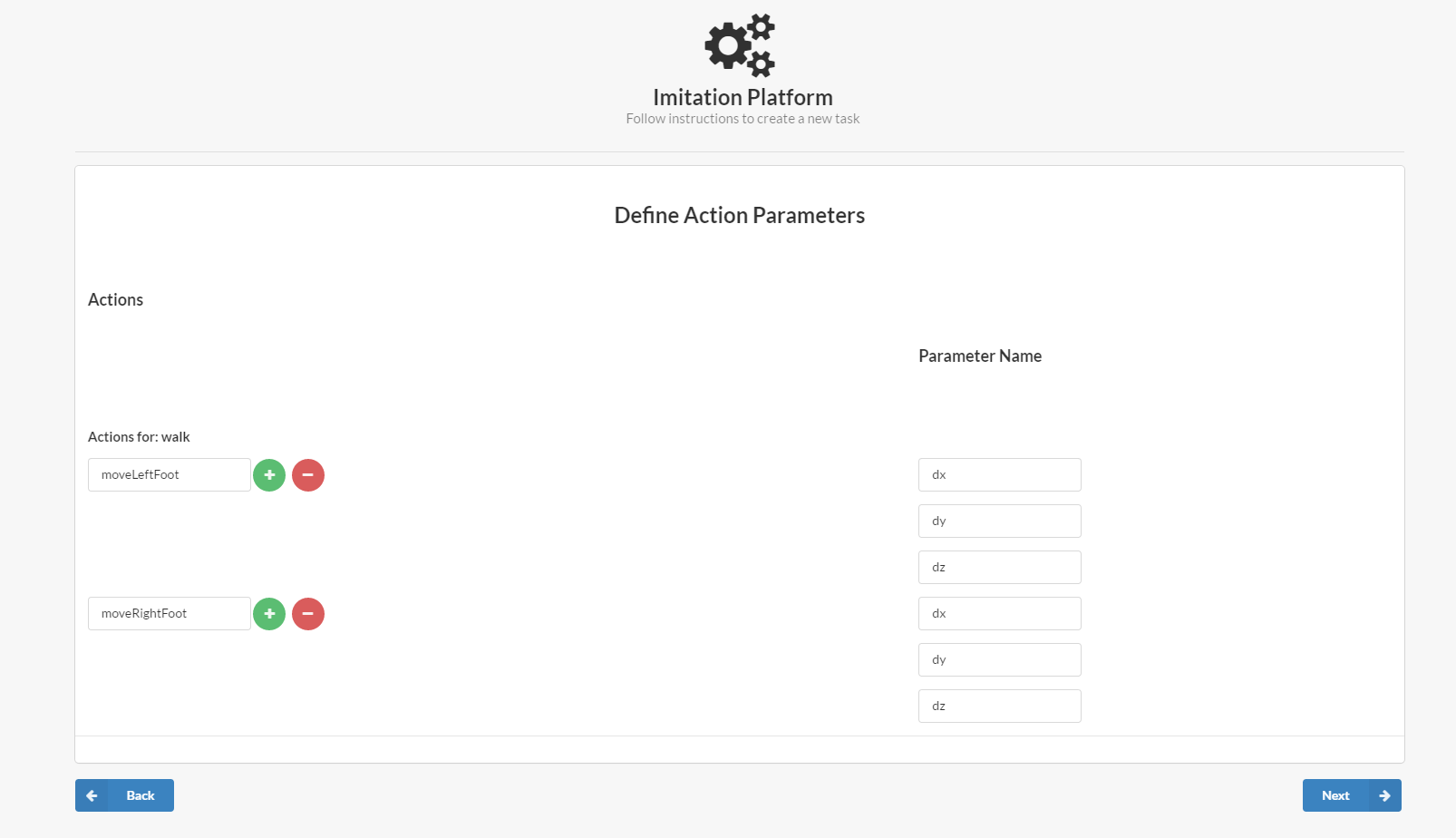


Figure 7. Define Action Parameters

1. Define Causal Parameters
   1. Define the parameters of the cause(s) that you defined in the first step. Naturally, all parameters of any given cause must be a subset of the parameters in its component actions.
   2. You may also toggle Custom Parameters to allow advanced input that is not explicitly a subset of the parameters of the component actions.

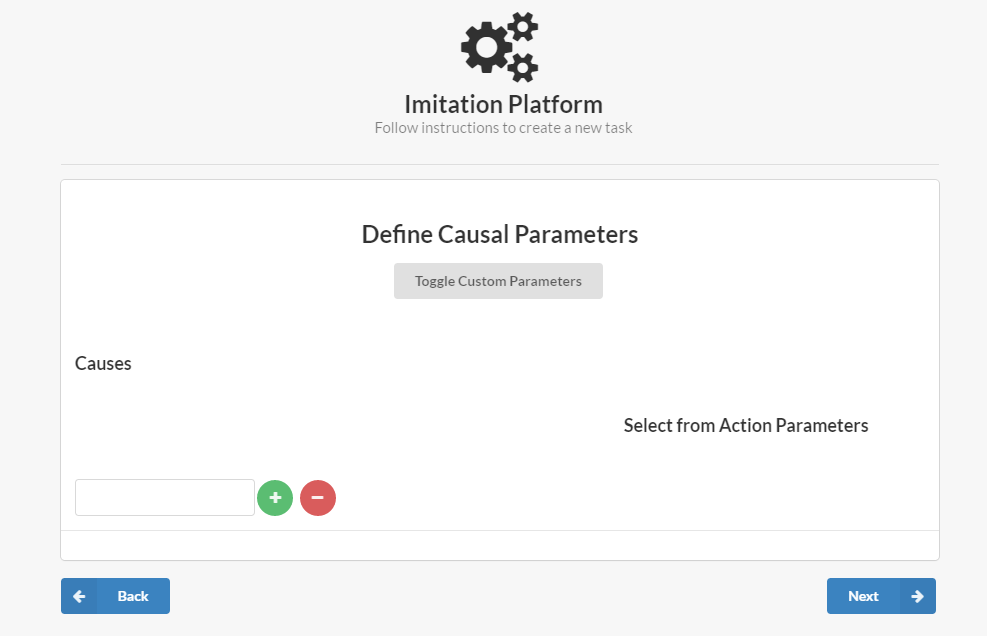


Figure 8. Define Causal Parameters

1. Click Next to complete the knowledge base. You should now save the knowledge base in a directory of your choice.

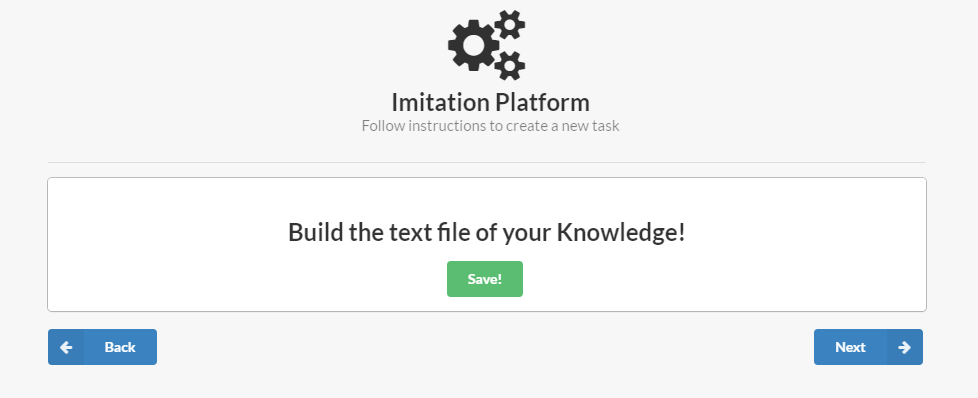


Figure . Save your custom knowledge base.

1. Click Next and add the causes and action file by opening the knowledge base you just created. Alternatively, you can go back into the knowledge base builder and create another one.

#### Insert Knowledge Base Script

The causal knowledge provided by the user with this option uses a custom syntax and should be implemented by advanced users only. Please see the documentation entitled **Causal Compiler Syntax User Guide.docx**. Once a knowledge base has been added with the Open File button, click Next.

# Insert Initial XML State

This step involves inserting the XML file which dictates the starting state of the environment. This is the environment that the imitation algorithm will manipulate in order to match the actions provided in the SMILE recording.

The XML may be provided in a number of ways. The two primary sources would be a direct export from a SMILE environment or as an output of the *Generate XML (Optional)* step as discussed above.

Once an initial state has been defined, click Run! Your output will be placed in “SMILE/tablesetup/” as “final\_xml.xml”.