This project involves developing a graphical user interface for our PyTorch Buildings library, a differentiable building energy simulation framework used for HVAC control research and building system optimization. Students will create a drag-and-drop interface similar to MATLAB Simulink or Modelica Dymola/OpenModelica tools, allowing researchers to visually construct building models by connecting components like thermal envelopes, rooftop units, VAV boxes, solar gain models, and intelligent control policies. The GUI should enable users to: (1) drag building components and control policies from a palette and place them on a canvas, (2) connect components by drawing wires between input/output ports (with automatic type checking), (3) set component parameters and control objectives through property dialogs, (4) configure and launch automated control policy training using our NeuroMANCER backend (students won't implement the training algorithms but will design the interface for specifying objectives like energy efficiency vs comfort trade-offs), (5) run simulations and visualize results with interactive plots showing both building performance and control behavior, and (6) save/load complete building system models with trained control policies. The underlying PyTorch Buildings library handles the physics simulation and NeuroMANCER handles the deep learning optimization - the GUI provides an intuitive visual layer for model construction, control design, and parameter management. Students will work with modern Python GUI frameworks (likely PyQt or Tkinter), learn about building physics and control concepts, and gain valuable experience with front-end applications for scientific machine learning workflows. The deliverable will be a **working prototype that enables researchers to design, train, and evaluate intelligent building control systems without writing code**.

On Monday we discussed PR and public/press release as well as started to outline the requirements document.

Your assignment is:

1. Create a Press Release
2. Outline Project Requirements

Please email your work to me by midnight, Monday, 17 November. (mowerj2@wwu.edu)

Press Release

In collaboration with Pacific Northwest National Laboratory, we are excited to announce the development of a new graphical user interface that simplifies the design, training, and evaluation of intelligent building energy systems.

This user-friendly tool allows researchers to visually create building models, optimize energy usage, and control HVAC systems.

The drag-and-drop interface enables users to select and place building components like thermal envelopes, HVAC units, and solar gain models. By connecting these components, users can easily configure system parameters, set control objectives, and run simulations to evaluate building performance.

Key Features Include:

- **Drag-and-Drop Interface**: Add building components and connect them visually.
- **Simulation & Results**: Run simulations and view interactive performance plots.
- **Control Policy Training**: Interface with the NeuroMANCER backend.
- **Save & Load Models**: Save and load building models and control policies for future use.

Expected Release

This PyTorch GUI tool will be available for use in late Spring of 2026 to help researchers optimize building plans for developing, and designing HVAC systems.

ADD MILESTONES

Project Requirements:

1. Project Overview

- **Goal**: Develop a graphical user interface for PyTorch Buildings, a framework for building energy simulation to design, simulate, and optimize building systems and HVAC controls.

2. Core Features

2.1 Drag-and-Drop Interface

- Users can drag building components such as thermal envelopes, HVAC systems and solar gain models onto a canvas and connect them.
- Each component has a properties dialog for setting parameters like energy efficiency goals, HVAC settings, etc.

2.2 Simulation, Visualization, and Control

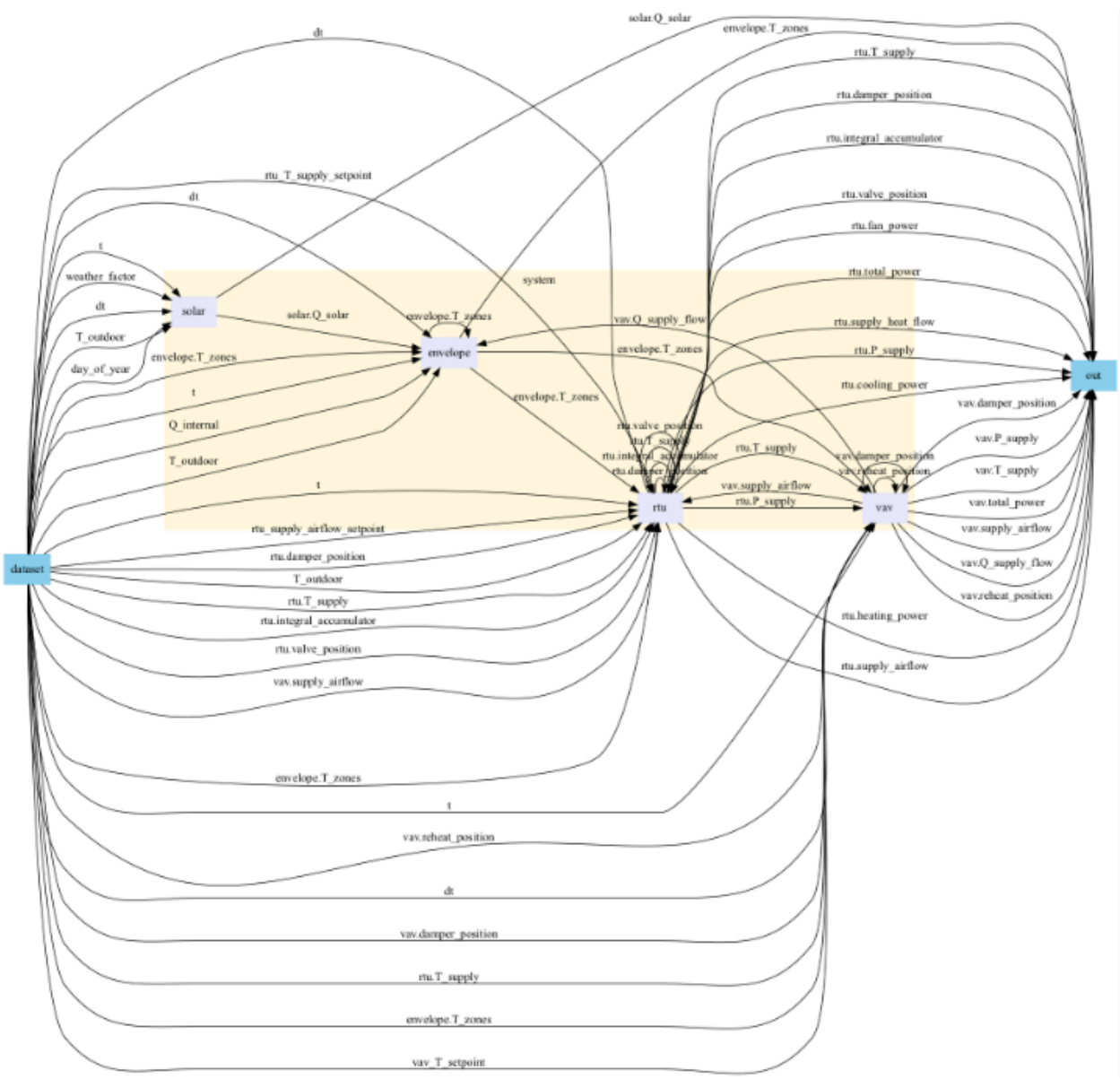
- Users can run simulations and view interactive plots to analyze building performance, energy use, and control policy effects.
- Interface with the NeuroMANCER backend to automatically train control policies that balance energy efficiency and comfort.

2.3 Model Save/Load

- Users can save and load entire system models, including trained control policies, for later use or analysis.

3. Technology Stack

- **Frontend**: PyQt or Tkinter for the GUI.
- **Backend**: PyTorch Buildings for physics simulations, NeuroMANCER for deep learning control policy training.


dataset
solar
envelope
rtu
vav
out
system
dt
t
weather_factor
T_outdoor
day_of_year
Q_internal
solar.Q_solar
envelope.T_zones
rtu_T_supply_setpoint
rtu_supply_airflow_setpoint
rtu.damper_position
rtu.T_supply
rtu.integral_accumulator
rtu.valve_position
vav.supply_airflow
rtu.fan_power
rtu.total_power
rtu.supply_heat_flow
rtu.P_supply
rtu.cooling_power
rtu.heating_power
vav.Q_supply_flow
vav.damper_position
vav.reheat_position
vav.P_supply
vav.T_supply
vav.total_power
vav.T_setpoint