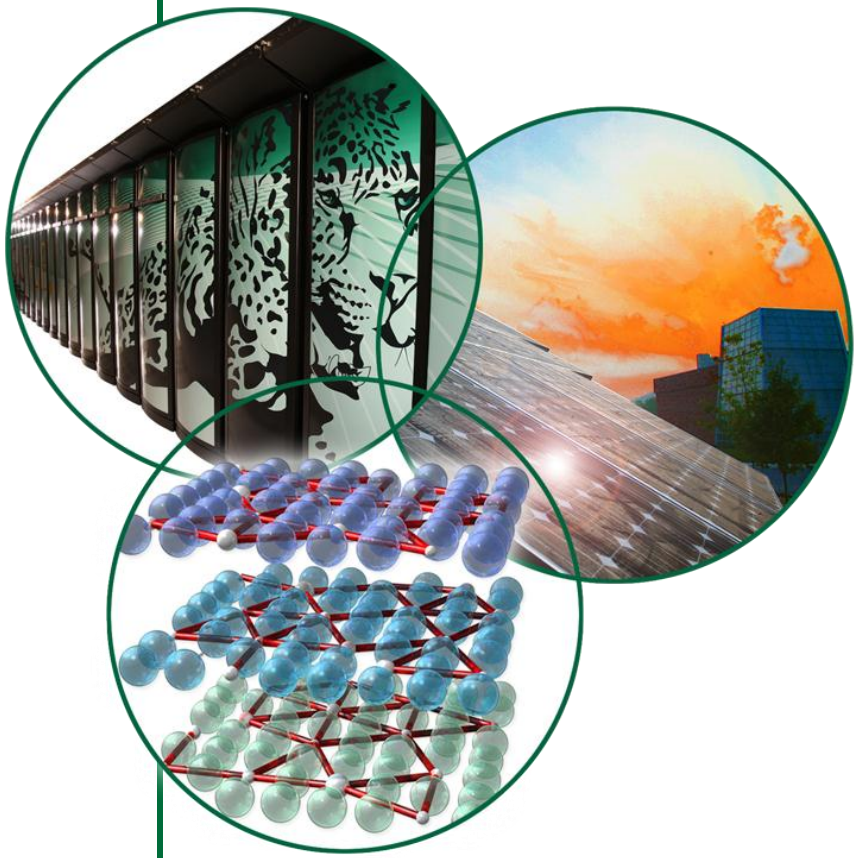
Adding Components to Open XAL

Christopher Allen
ORNL



U.S. DEPARTMENT OF ENERGY

OAK RIDGE NATIONAL LABORATORY
MANAGED BY UT-BATTELLE FOR THE DEPARTMENT OF ENERGY

Outline

- Machine Representation vs. Machine Simulation
- Representing New Hardware
- Simulating New Hardware

Hardware Type vs. Modeling Types

- Within XAL there are classes that
  - represent hardware (SMF)
  - simulate hardware (Online Model)

- In order to introduce new hardware into XAL, you must first create a hardware representation of it.

- If you wish to simulate the effects of the new hardware, you must implement a model of it.

- Finally, you must bind the hardware type into XAL, then (optionally) bind the model element to the hardware element.

Creating New Hardware Types (SMF)

Steps

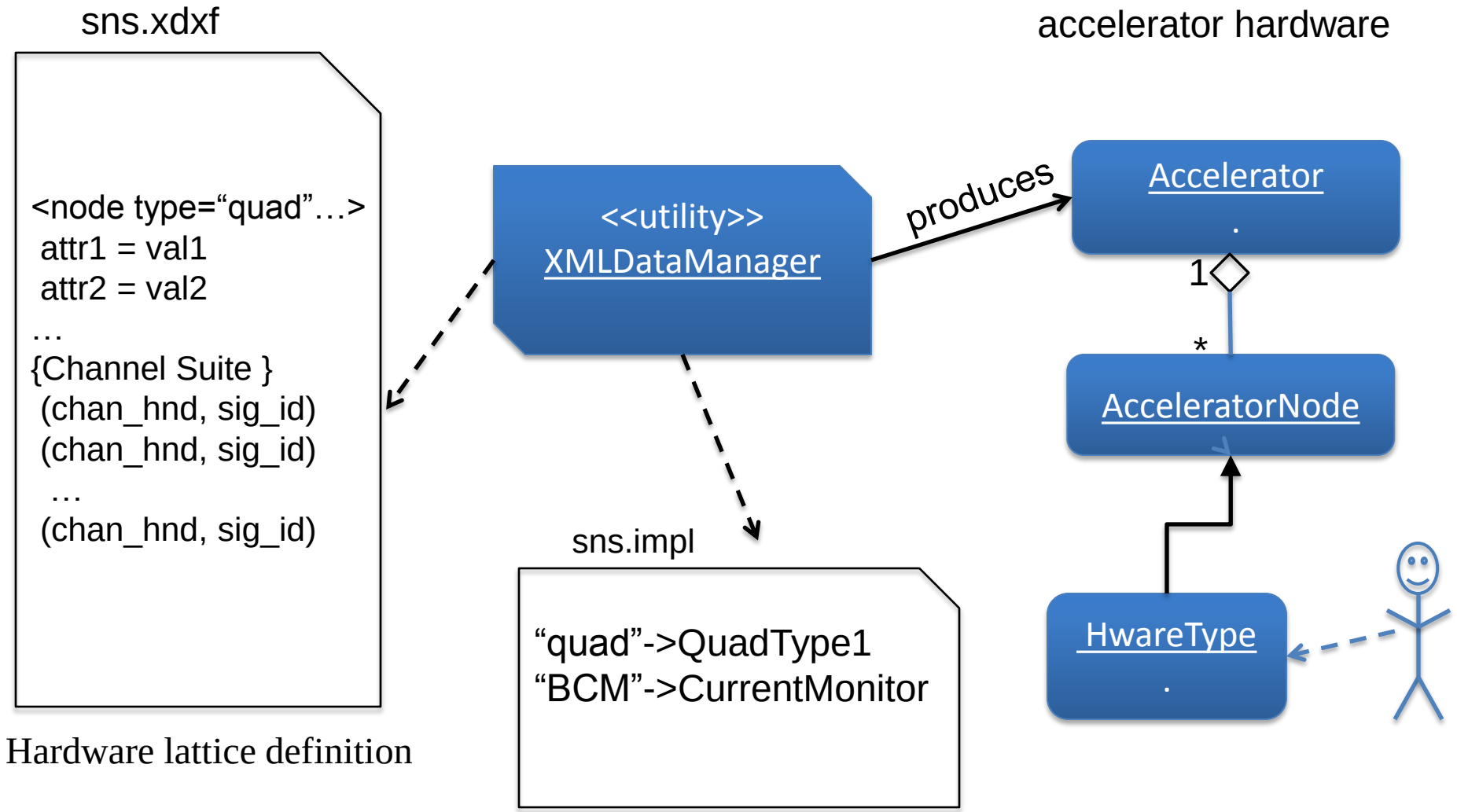
1. Implement the new class type with an *unique type ID* string
   - Derived from xal.smf.AcceleratorNode base class
   - Contains support for all the hardware channels it needs (see next)

2. Register new class in `sns.impl` with type ID string as key

3. Create the ChannelSuite for hardware type in `sns.xdxf`
   - Maps the generic "handles" used by hardware classes to channel names of the actual hardware devices
   - (Hardware instances of the same type have the same handles – however, they are bound to different channels.)
   - ChannelSuite is a registry of all hardware connections for the hardware type.

Hardware Representation

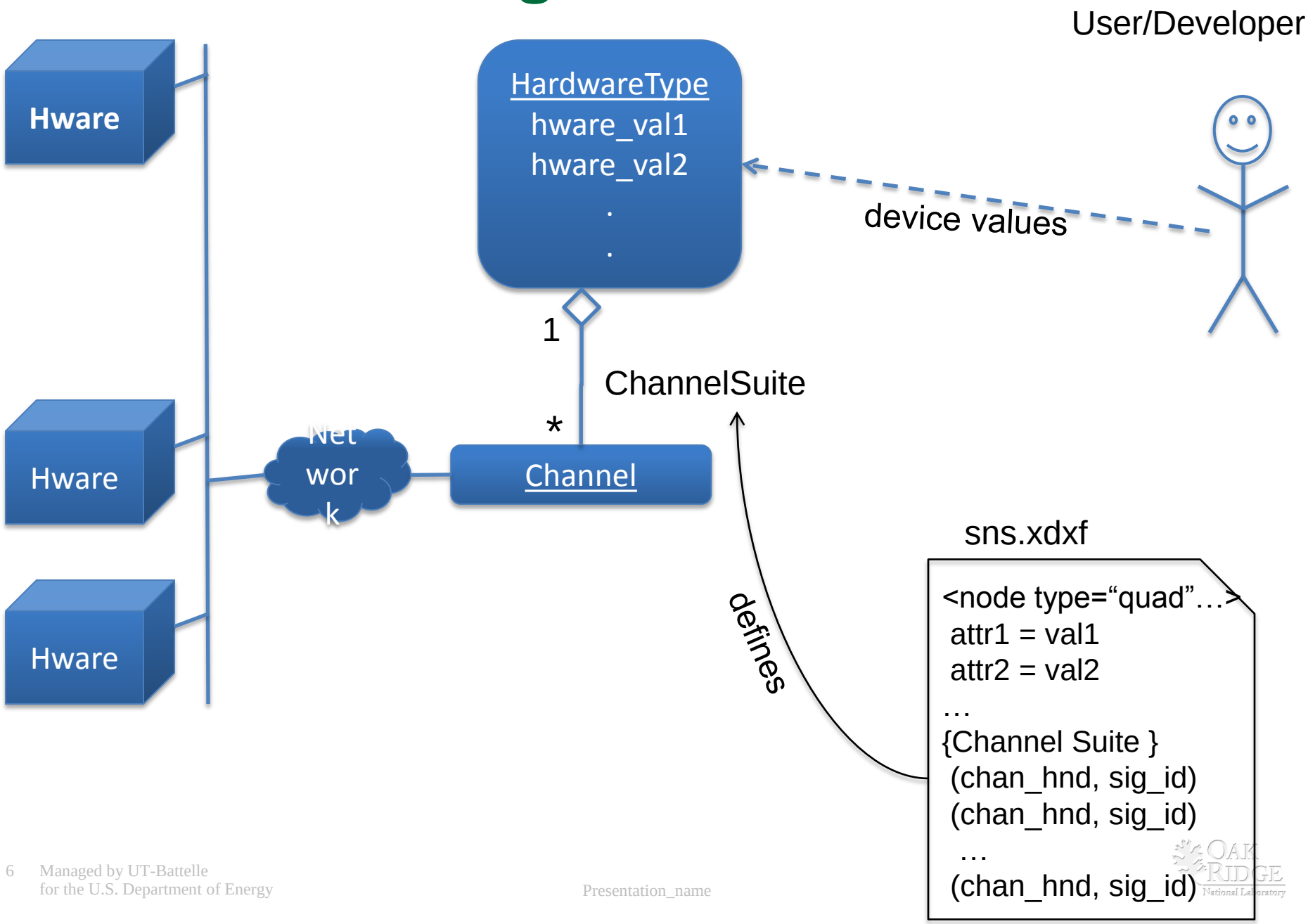
SMF data structure: developer interface to accelerator hardware



Hardware lattice definition

Device type id to device class name binding

Hardware Bindings



Simulating New Hardware

Steps

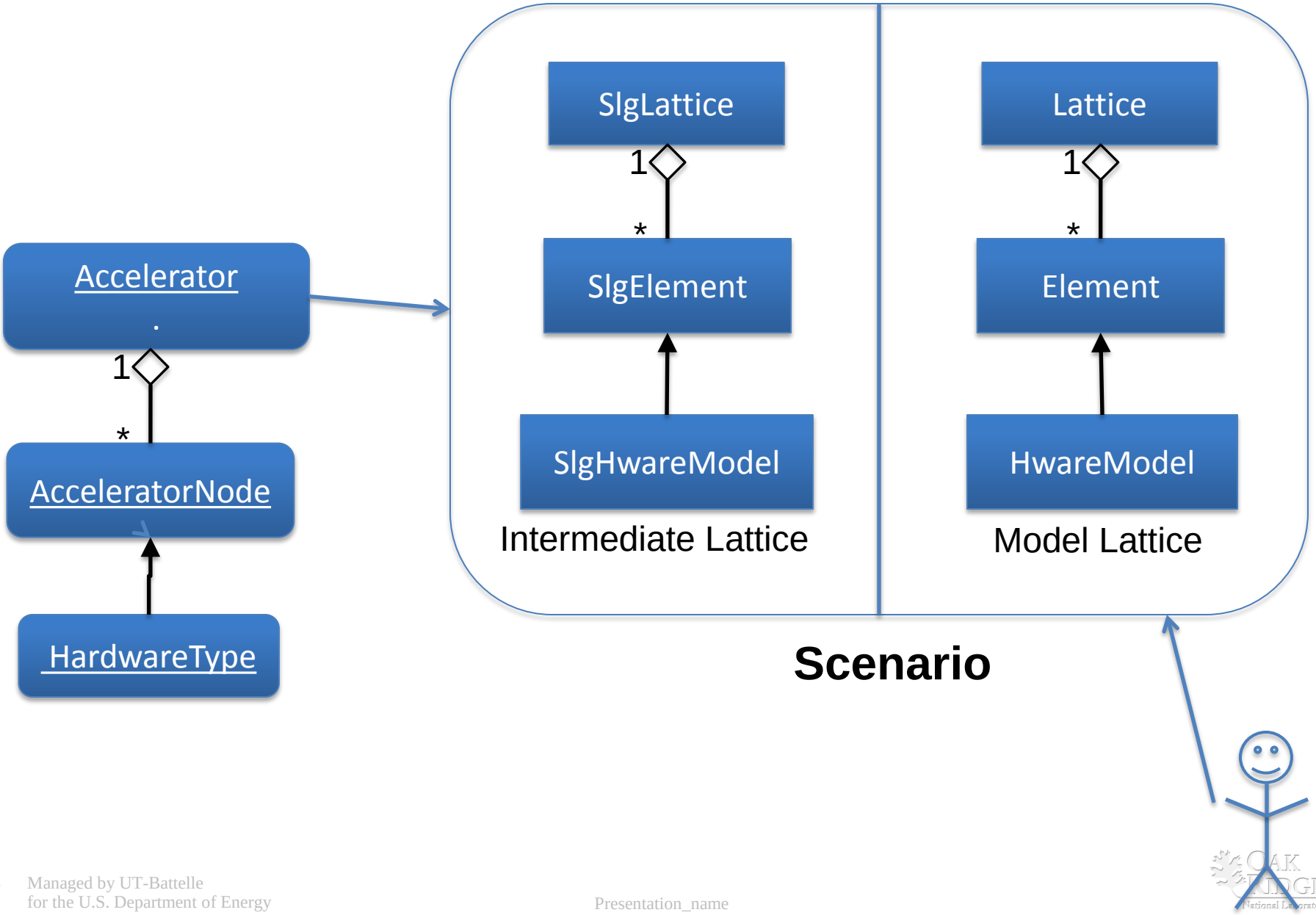
1. Implement a new modeling element derived from the *Element* class.
   - Must implement the *transferMap()* method.

2. *Add a hook in the lattice generator (SLG) to bind the hardware type (SMF) to the new modeling element.

3. Within the intermediate lattice (SLG) initialize the modeling element
   - Set all the static properties (e.g., length, design field, etc.)
   - Create all the synchronization for the dynamic properties (e.g., fields)

*Unfortunately this can be a convoluted process. We have a new design for the lattice generator which would alleviate most of the current shortcomings.

Model Lattice Generation



Summary

- Adding New Hardware
  - Relatively Easy
    - Derive new hardware from *AcceleratorNode* and register in *sns.impl*
- Adding New Modeling Elements
  - Might be difficult due to intermediate lattice
    - Replacement of an element is straightforward
    - Creating a whole new hardware/modeling element pair is involved
  - Once the new lattice generator is implemented the addition of new modeling elements will be similar to that of hardware elements