

2025 ASHRAE WINTER CONFERENCE

ORLANDO, FEB 8-12 | AHR EXPO, FEB 10-12

Seminar 24

**Livestream: The "Killer Application" How new ASHRAE Standards can
Unleash Building Control Efficiency and Grid Support**

**Future Tools and Solutions for Brownfield
Deployment of Building Ontologies**

Dr. Gabe Fierro

Colorado School of Mines

Learning Objectives

Learning Objectives:

1. Describe what new standards enable and why this is so important for the future of the built environment.
2. Learn about practical examples of the implementation of these standards in real buildings.
- 3. Understand how new open-source tools can facilitate the adoption of these new standards in existing buildings.**
- 4. Explore opportunities to use these standards in your products, services or workflows.**

ASHRAE is a Registered Provider with the American Institute of Architects Continuing Education Systems. Credit earned on completion of this program will be reported to ASHRAE Records for AIA members. Certificates of Completion for non-AIA members are available on request.

This program is registered with the AIA/ASHRAE for continuing professional education. As such, it does not include content that may be deemed or construed to be an approval or endorsement by the AIA or any material of construction or any method or manner of handling, using, distributing, or dealing in any material or product. Questions related to specific materials, methods, and services will be addressed at the conclusion of this presentation.

Acknowledgements

- Avijit Saha, NREL
- Tobias Shapinsky, NREL
- Matthew Steen, NREL
- Hannah Eslinger, NREL
- Lazlo Paul, LBNL
- Marco Pritoni, LBNL
- Paul Ehrlich, Building Intelligence Group
- ASHRAE S223P Committee
- ASHRAE S231P Committee
- Amir Roth, DOE
- Cecilia Johnson, DOE
- This work was supported by the Assistant Secretary for Energy Efficiency and Renewable Energy , Building Technologies Office, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

Outline

- Challenges in brownfield deployment
- Brownfield metadata sources
- 223P model prototyping with point labels
- 223P model validation for application configuration
- Conclusions

Overview

- Our goal is to **automatically** create **useful** and **valid** 223P models which support applications like 231P control sequences
- **(Semi-)Automatically**: minimizing manual data entry and configuration
- **Useful**: the 223P model contains enough metadata to support applications
- **Valid**: the model is a faithful and accurate representation of the building and its assets
- *How?*
 - Use point labels and point/equip schedules
 - Produce 223P models and shapes

Overview

Metadata Sources

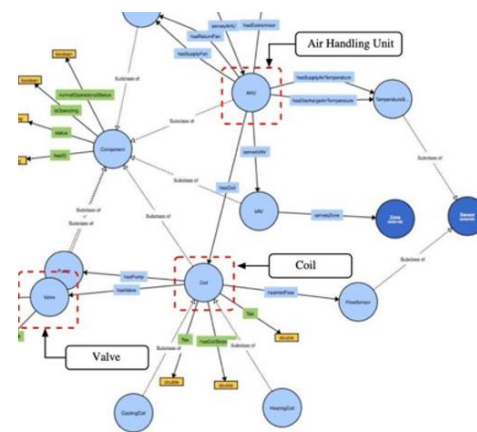
label
:BuildingName_02:FCU503_ChwVlvPos
:BuildingName_01:FCU336_OccHtgSptFnl
:BuildingName_02:FCU510_EffOcc
:BuildingName_02:FCU507_UnocHtgSpt
:BuildingName_02:FCU415_UnocHtgSpt
:BuildingName_01:FCU203_OccClgSpt
:BuildingName_02:FCU521_UO11_HwVlvOut
:BuildingName_01:FCU365_UnocHtgSptFnl
:BuildingName_02:FCU529_UnocHtgSpt

Point Labels

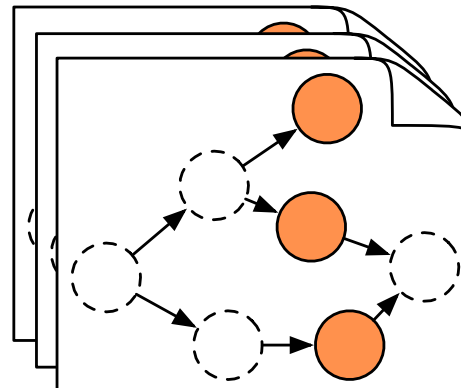
DESCRIPTION	I/O POINTS			
	AI	DI	AO	DO
Package Units 2				
Return Air Temp/RH	4			
Cooling Coil off Temp	2			
Heater coil off Temp	2			
Room Air Temp/RH	8			
Supply fan run status		2		
Supply fan trip status		2		
Filter status		2		
Stage -01 Run status		2		
Stage -02 Run Status		2		
Stage -03 Run status		2		

Point and Equipment Schedules

Model Development

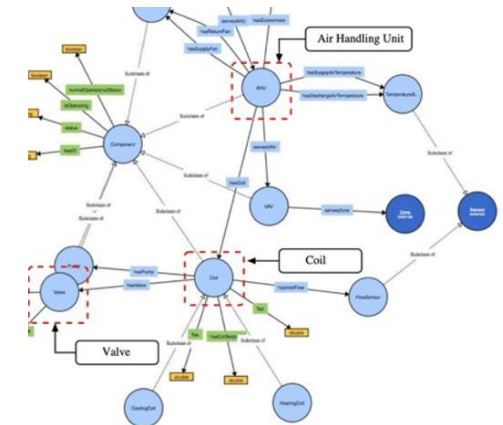


223P Model

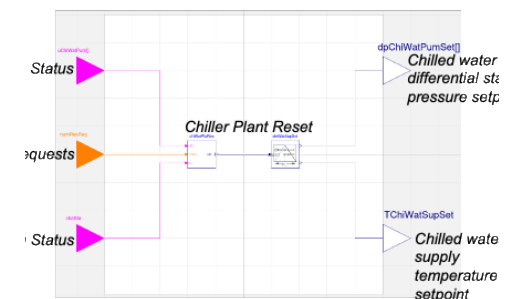


Validation Shapes

Outputs



Validated 223P Model

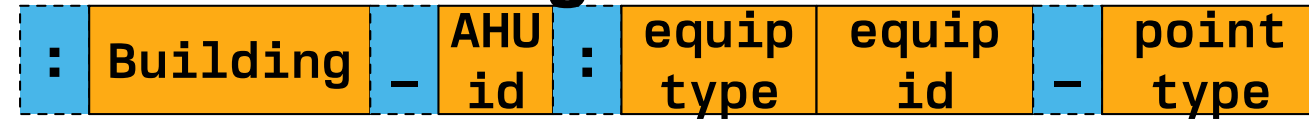


Configured Control Sequences and FDD Rules

Point Labels

- Use the structure of point labels to extract building metadata
- Define the **naming convention** using simple code
- Extracts entities and their 223P types from each label

Naming Convention



label
:BuildingName_02:FCU503_ChwVlvPos
:BuildingName_01:FCU336_OccHtgSptFnl
:BuildingName_02:FCU510_EffOcc
:BuildingName_02:FCU507_UnoccHtgSpt
:BuildingName_02:FCU415_UnoccHtgSpt
:BuildingName_01:FCU203_OccClgSpt
:BuildingName_02:FCU521_UO11_HwVlvOut
:BuildingName_01:FCU365_UnoccHtgSptFnl
:BuildingName_02:FCU529_UnoccHtgSpt

Point Labels

- Use the structure of point labels to extract building metadata
- Define the **naming convention** using simple code
- Extracts entities and their 223P types from each label
 - **“Entity”**: an instance of some 223P-defined class

Naming Convention

```
sequence(  
  : string(":", Delimiter),  
    #### Building  
  Building constant(Constant(BRICK.Building)),  
    regex(r"^[^_]+", Identifier),  
  : string("_", Delimiter),  
    #### AHU  
  AHU id constant(Constant(S223.AirHandlingUnit)),  
    regex(r"[0-9a-zA-Z]+", Identifier),  
  : string(":", Delimiter),  
    #### equipment type and identifier  
  equip type equip id equip_abbreviations,  
    regex(r"[0-9a-zA-Z]+", Identifier),  
  : string("_", Delimiter),  
    #### point types  
  point type point_abbreviations,  
  )
```


Point Labels

```

sequence(
  string(":", Delimiter),
  ##### Building
  constant(Constant(BRICK.Building)),
  regex(r"^[_]+", Identifier),
  string("_", Delimiter),
  ##### AHU
  constant(Constant(S223.AirHandlingUnit)),
  regex(r"[0-9a-zA-Z]+", Identifier),
  string(":", Delimiter),
  ##### equipment type and identifier
  equip_abbreviations,
  regex(r"[0-9a-zA-Z]+", Identifier),
  string("_", Delimiter),
  ##### point types
  point_abbreviations,
)

```

Building

AHU id

equip type

equip id

point type

Naming convention
parses labels



label
:BuildingName_02:FCU503_ChwVlvPos
:BuildingName_01:FCU336_OccHtgSptFnl
:BuildingName_02:FCU510_EffOcc

Outputs entities
with 223P classes



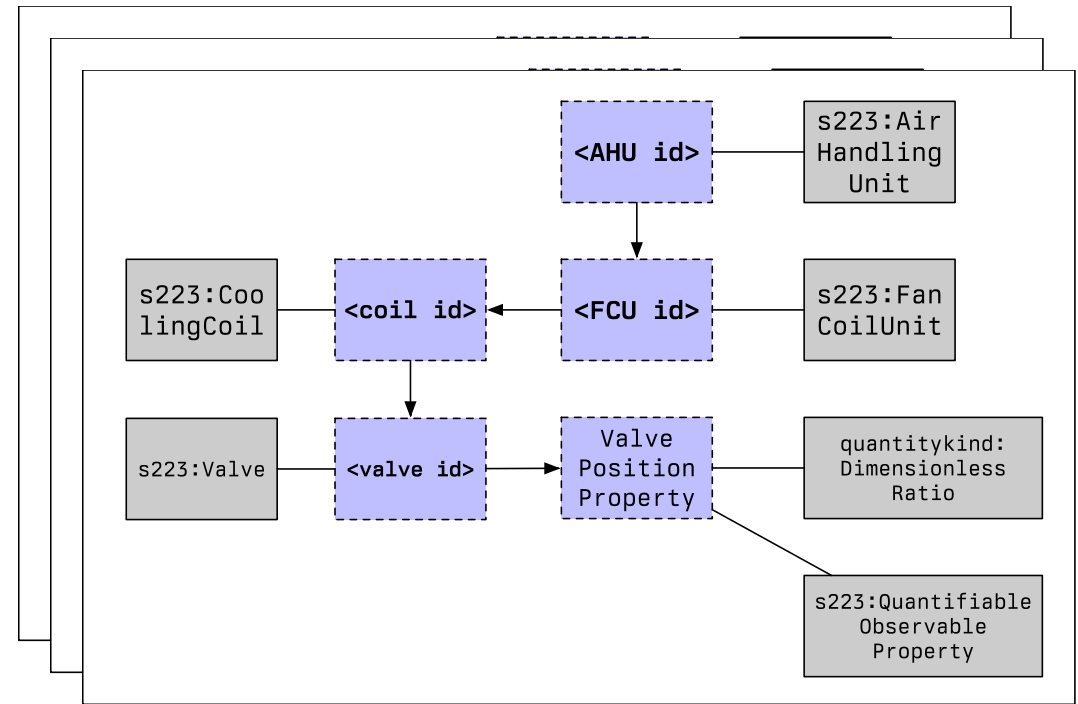
:	Building	-	AHU id	:	equip type	equip id	-	point type
:	BuildingName	_	02	:	FCU	503	-	Chilled Water Valve Position
:	BuildingName	_	01	:	FCU	336	-	Occupied Heating Setpoint
:	BuildingName	_	02	:	FCU	510	-	Effective Occupancy Status

Generating 223P Models with Point Labels

- Next, combine entities into a 223P model
- Two ways to build the model:

```
class s223 Equipment extends s223 Connectable
{
  * s223 actuatedByProperty s223 ActuatableProperty;
  * s223 connectedFrom ()
  * s223 connectedTo ()
  * s223 contains ( s223 Equipment | s223 Junction )
  * s223 executes s223 Function;
  * s223 hasPhysicalLocation s223 PhysicalSpace;
  * s223 hasRole s223 EnumerationKind-Role;
}
```

Use 223P ontology definitions to suggest relationships

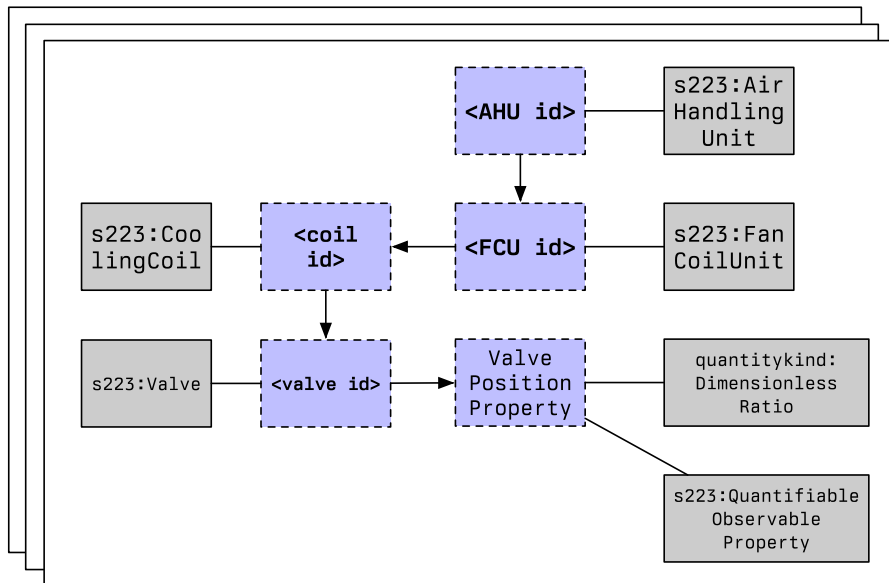


Use templates of common system configurations

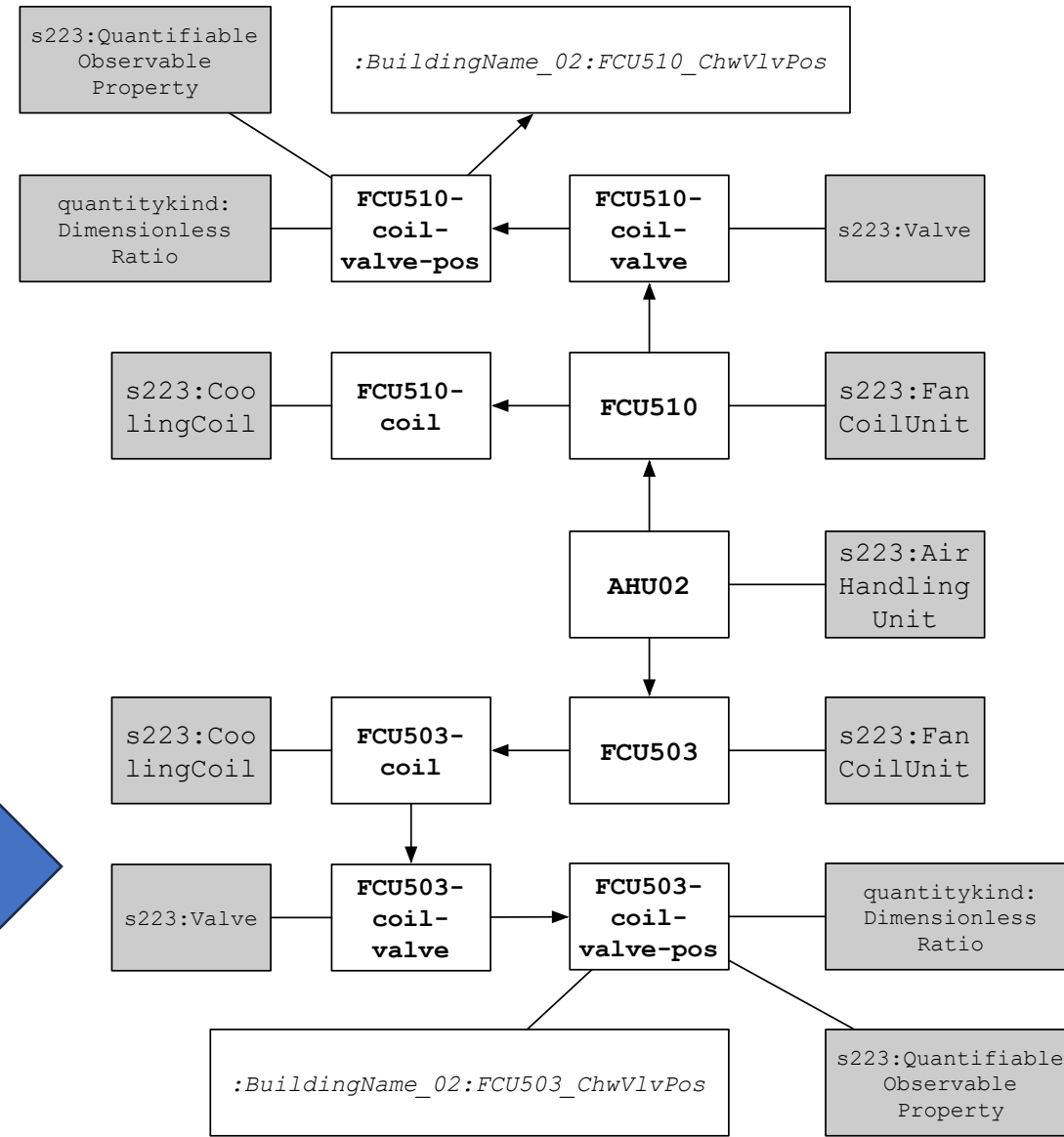
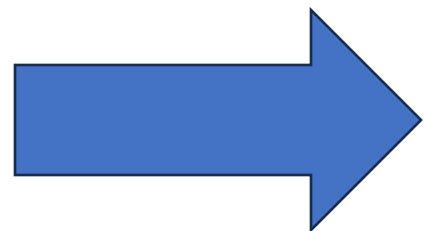
Generating 223P Models with Point Labels

Building	AHU id	equip type	equip id	point type
BuildingName_02	02	FCU	503	Chilled Water Valve Position
BuildingName_01	01	FCU	336	Occupied Heating Setpoint
BuildingName_02	02	FCU	510	Effective Occupancy Status

Find and apply templates



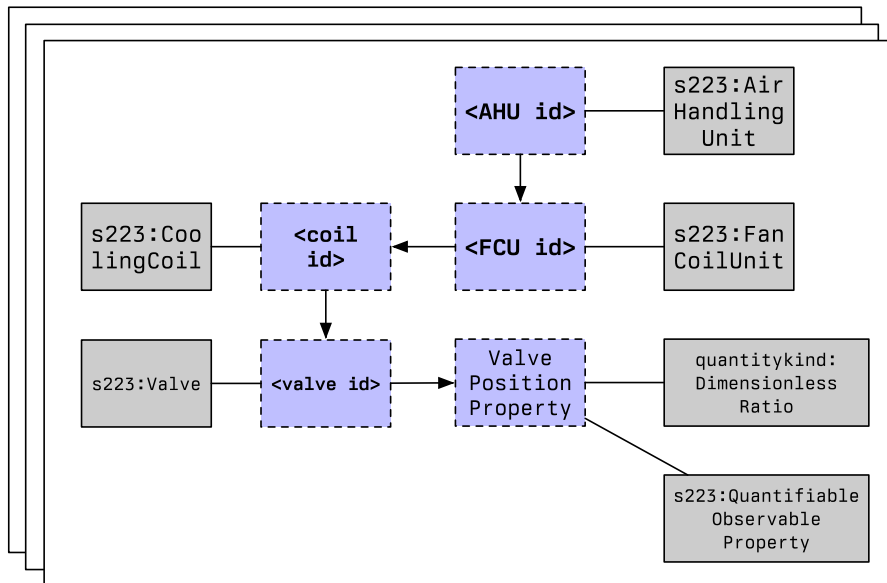
Generate 223P Model



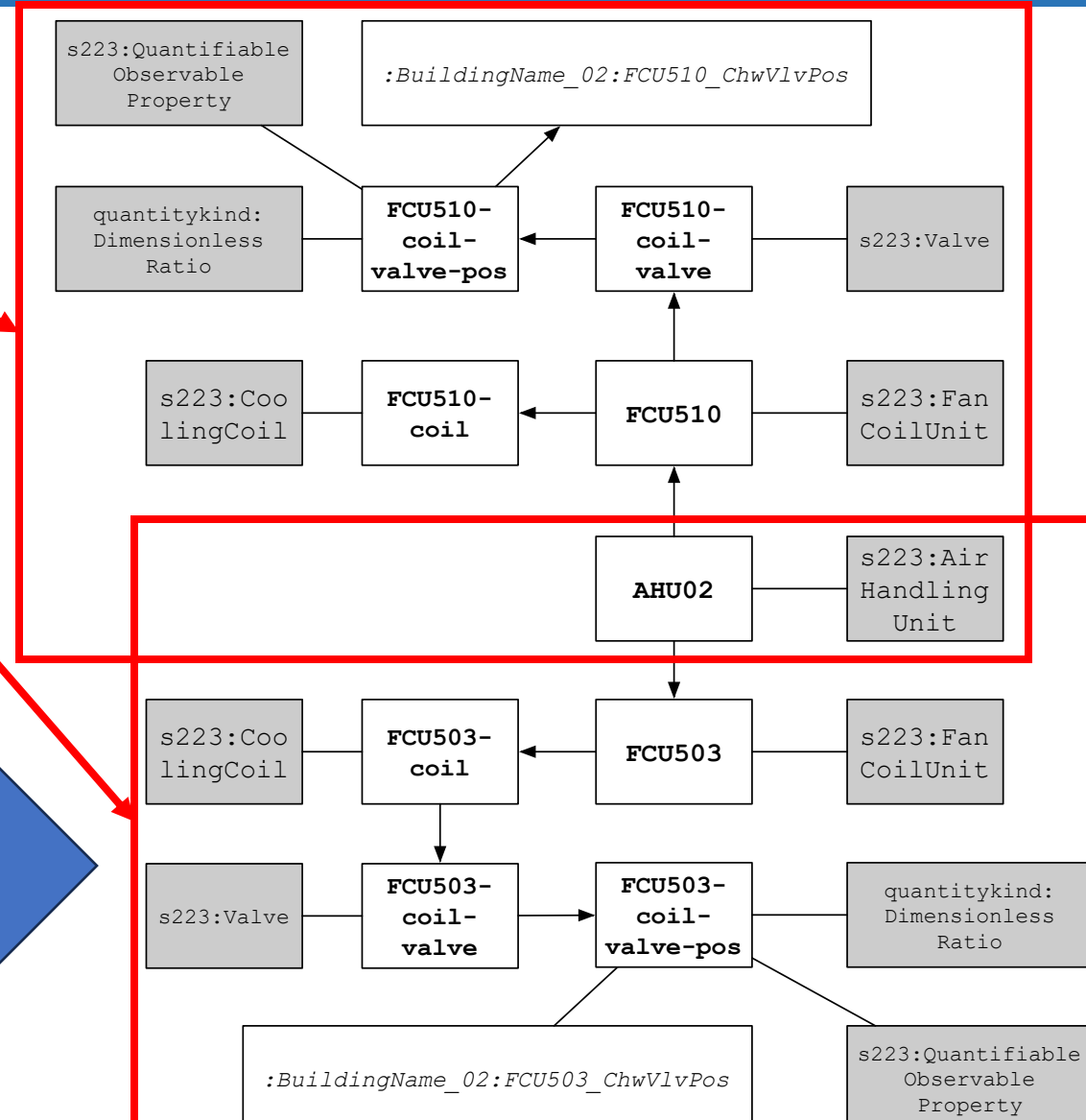
Generating 223P Models with Point Labels

Building	AHU id	equip type	equip id	point type
BuildingName_02	02	FCU	503	Chilled Water Valve Position
BuildingName_01	01	FCU	336	Occupied Heating Setpoint
BuildingName_02	02	FCU	510	Effective Occupancy Status

Find and apply templates



Generate 223P Model



Point and Equipment Schedules

- How to verify the model?
- **Point schedules** define which types of points to expect on which kinds of equipment
- **Equipment schedules** define how many of each equipment we should see in the model
- Convert schedules to **shapes** for validating the model

DESCRIPTION	I/O POINTS			
	AI	DI	AO	DO
Package Units 2				
Return Air Temp/RH	4			
Cooling Coil off Temp	2			
Heater coil off Temp	2			
Room Air Temp/RH	8			
Supply fan run status		2		
Supply fan trip status		2		
Filter status		2		

Schedule



Terminal Units must have:

Points

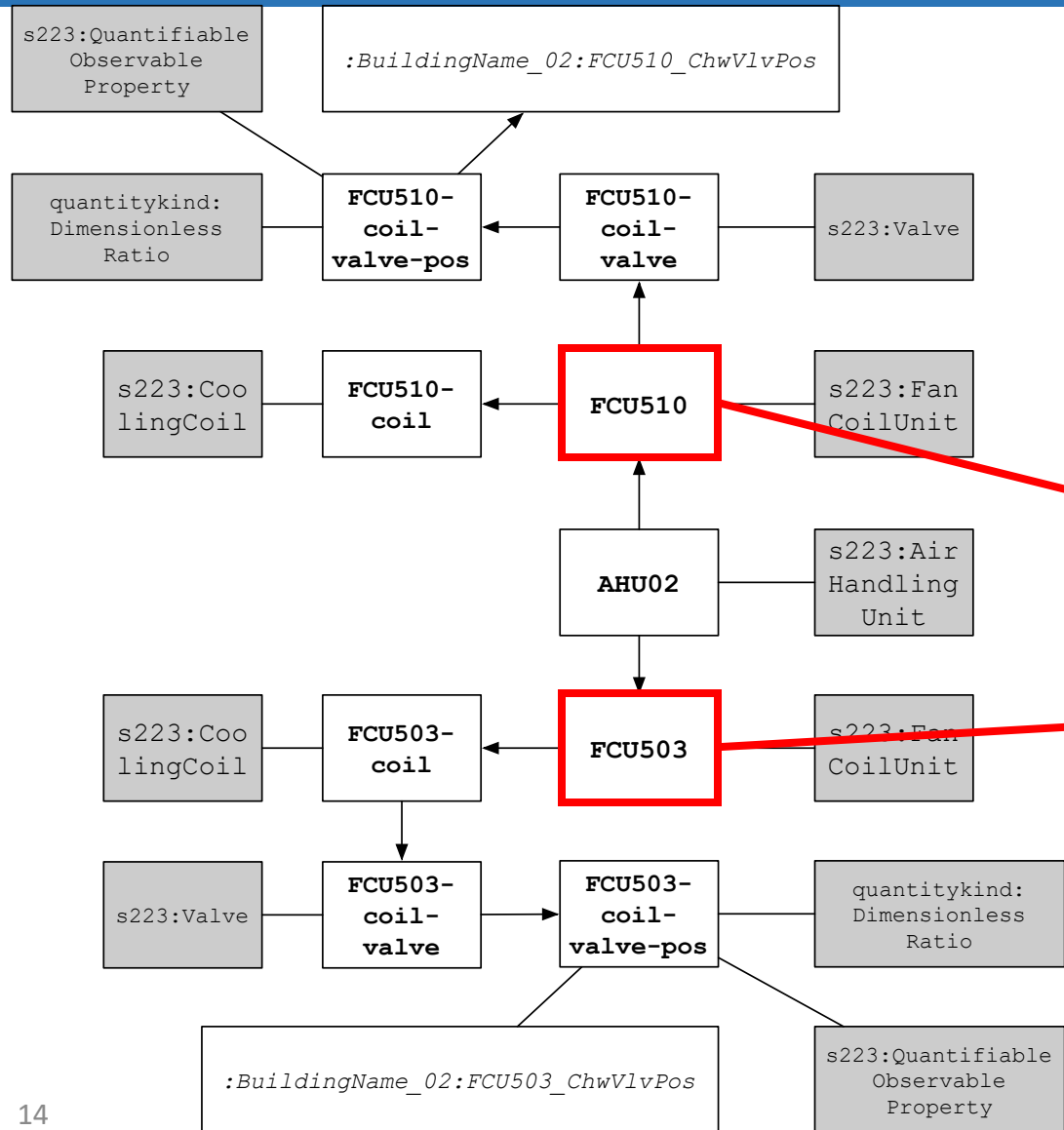
- Return Air Temperature Sensor
- Return Air Relative Humidity Sensor
- Room Air Temperature Sensor
- ...

Parts

- Cooling Coil
- Heating Coil
- Filter
- ...

Shape

Using Shapes to Validate a Model



Terminal Units must have:

Points

- Return Air Temperature Sensor
- Return Air Relative Humidity Sensor
- Room Air Temperature Sensor
- ...

Parts

- Cooling Coil
- Heating Coil
- Filter
- ...

Schedule-derived shapes



SHACL Validation Process

Shapes validate 223P model content



FCU503	Passes Validation
FCU510	FAILS Validation: <ul style="list-style-type: none"> - missing Room Air Temp Sensor - missing Heating Coil - missing Filter

Validation Report

Using Validation to Fix a Model

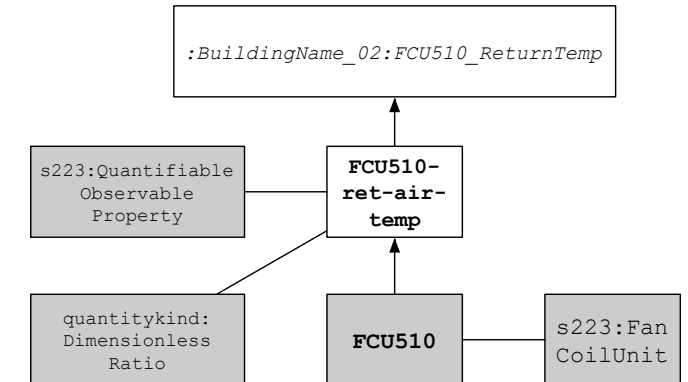
From validation process



FCU503	Passes Validation
FCU510	FAILS Validation: <ul style="list-style-type: none">- missing Room Air Temp Sensor- missing Heating Coil- missing Filter

Validation Report

To validation process

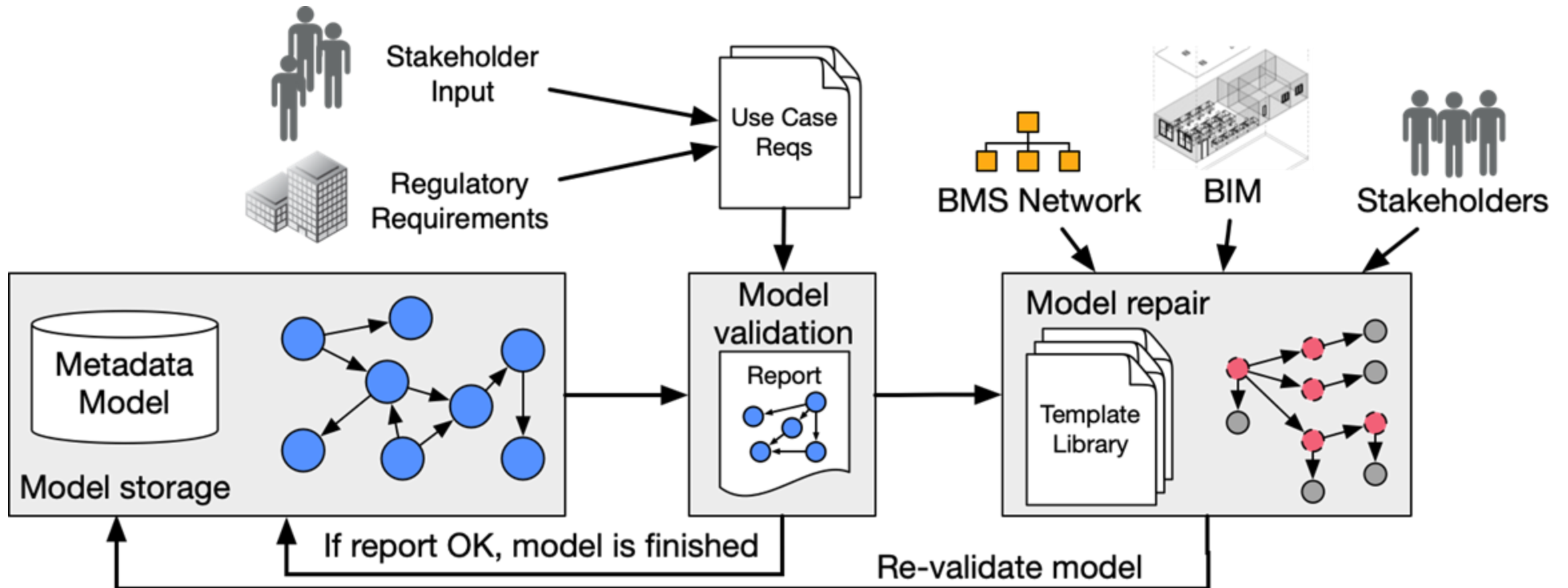


Model Patch

Possible fixes

- Extend point label naming convention script
- Re-scan network for more points
- Extract metadata from BIM
- Prompt user for specific inputs

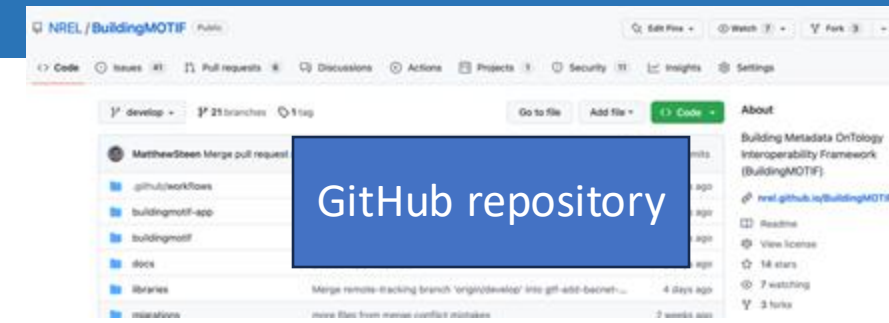
Putting It All Together



Combining both **model generation/repair** and **model validation** into an iterative workflow

Open-Source Tool

- **BuildingMOTIF software development kit**
 - Available on GitHub under NREL organization
 - Released under permissive BSD 3 Clause license
 - Tutorials, notebooks, demo web interface
 - Supports Brick, 223P and soon Haystack
 - *Simplifies building metadata-driven software*



Application-Driven Creation of Building Metadata Models with Semantic Sufficiency

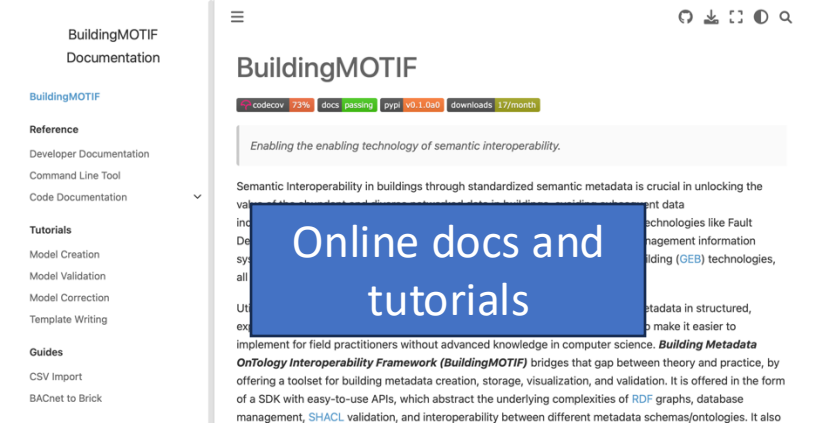
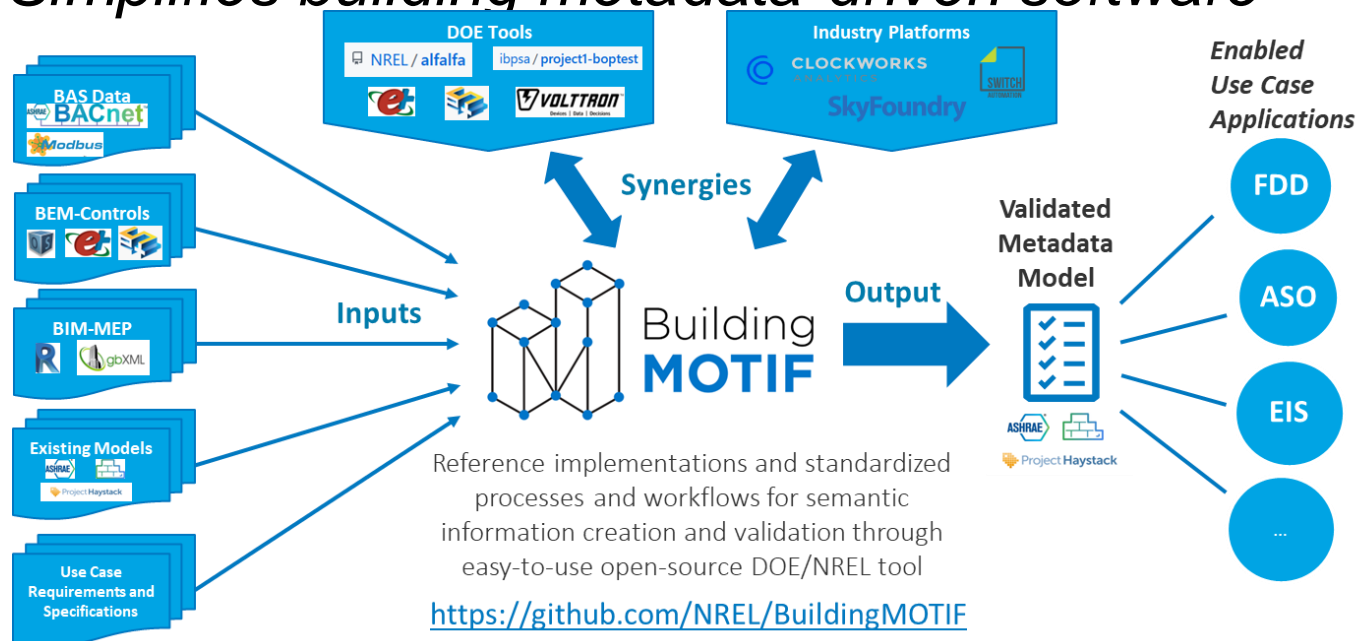
Gabe Fierro
gtfierro@mines.edu
Colorado School of Mines
National Renewable Energy
Laboratory
Golden, Colorado, U.S.A.

Avijit Saha
Avijit.Saha@nrel.gov
National Renewable Energy
Laboratory
Golden, Colorado, U.S.A.

Tobias Shapinsky
Tobias.Shapinsky@nrel.gov
National Renewable Energy
Laboratory
Golden, Colorado, U.S.A.

Research papers
on internals

ABSTRACT
Semantic metadata models such as Brick, Haystack, and BOT promise to simplify and lower the cost of developing software for smart buildings, enabling the widespread deployment of energy efficiency applications. However, creating these models remains a challenge. Despite recent advances in creating models from existing digital representations like point labels...



Conclusion

- **Shapes** are specifications of model requirements
 - Point and equipment schedules are useful for building the model
 - 231P control sequences can be packaged as shapes
 - Shapes ensure models have enough information to configure applications like FDD and control sequences
- **Open-source tools** can build 223P models by parsing point labels
 - Other sources of metadata input are available
 - Spreadsheets, web forms, etc
 - *Create your own tools on top!*
- **Validation** ensure that 223P models contain correct and sufficient information
 - Use this to (semi-)automatically configure 231P sequences, FDD, etc
- These efforts get better with data! Reach out to us if you have point labels to donate

Questions

Dr. Gabe Fierro

gtfierro@mines.edu