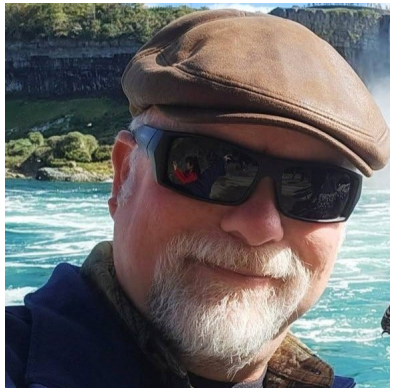

NIAGARA
SUMMIT
2026
SEAMLESS CONNECTIVITY,
POWERFUL INTELLIGENCE
TRIDIUM

Data-Driven Success : Leveraging Data Modeling



James Johnson

Eric Anderson

TRIDIUM

This document is a non-binding, confidential document that contains valuable proprietary and confidential information of Tridium and must not be disclosed to any third party without our written agreement. It does not create any binding obligations on us to develop or sell any product, service or offering. Content provided herein cannot be altered or modified and must remain in the format as originally presented by Tridium. Any descriptions of future product direction, intended updates, or new or improved features or functions are intended for informational purposes only and are not binding commitments on us and the sale, development, release, or timing of any such products, updates, features or functions is at our sole discretion All demos included in this presentation are for informational purposes only.

TRIDIUM

Agenda

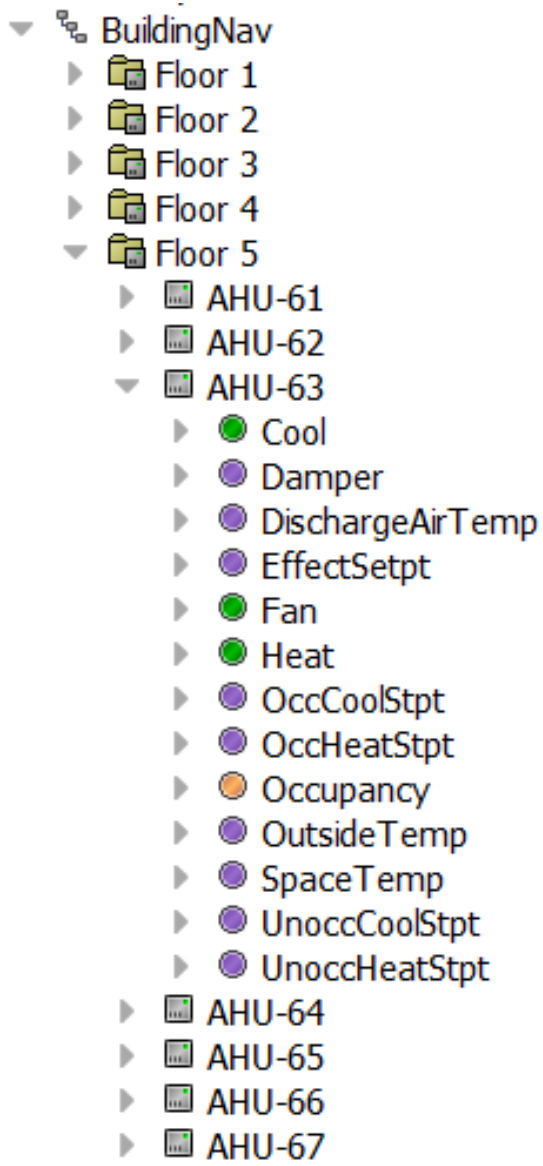
- What is the value proposition?
- Which modeling standard? Haystack? Brick?
- How to add tags and relations efficiently

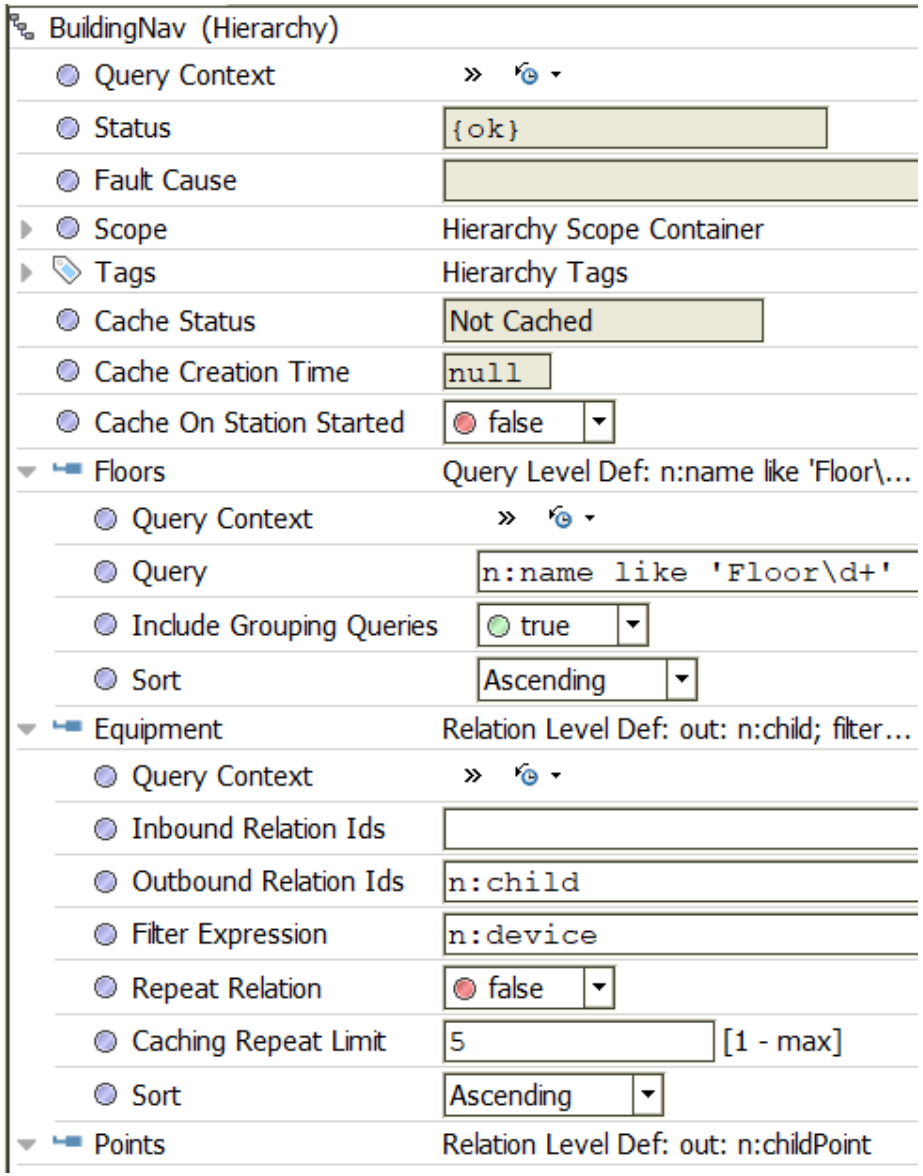
Hierarchy Use Cases

Hierarchies – Flexible Navigation

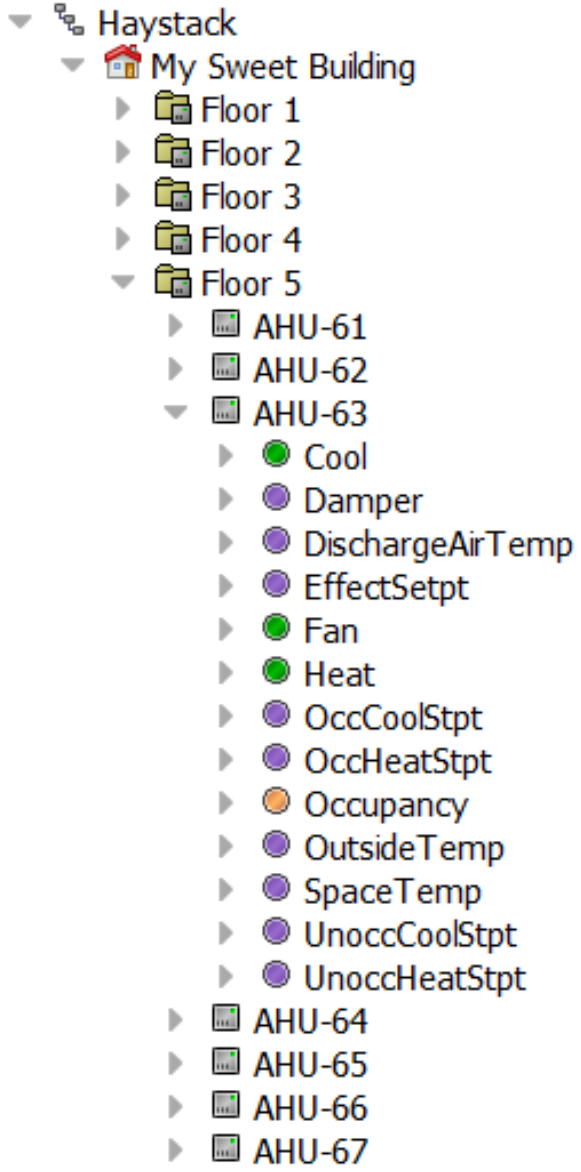
- Relations can be utilized to develop functional navigation based on physical layout in the building, systems, schedules, etc.
- Multiple navigation hierarchies can be assigned to the same user.
- Provides history grouping for Web Charts, Niagara Analytics and other features.

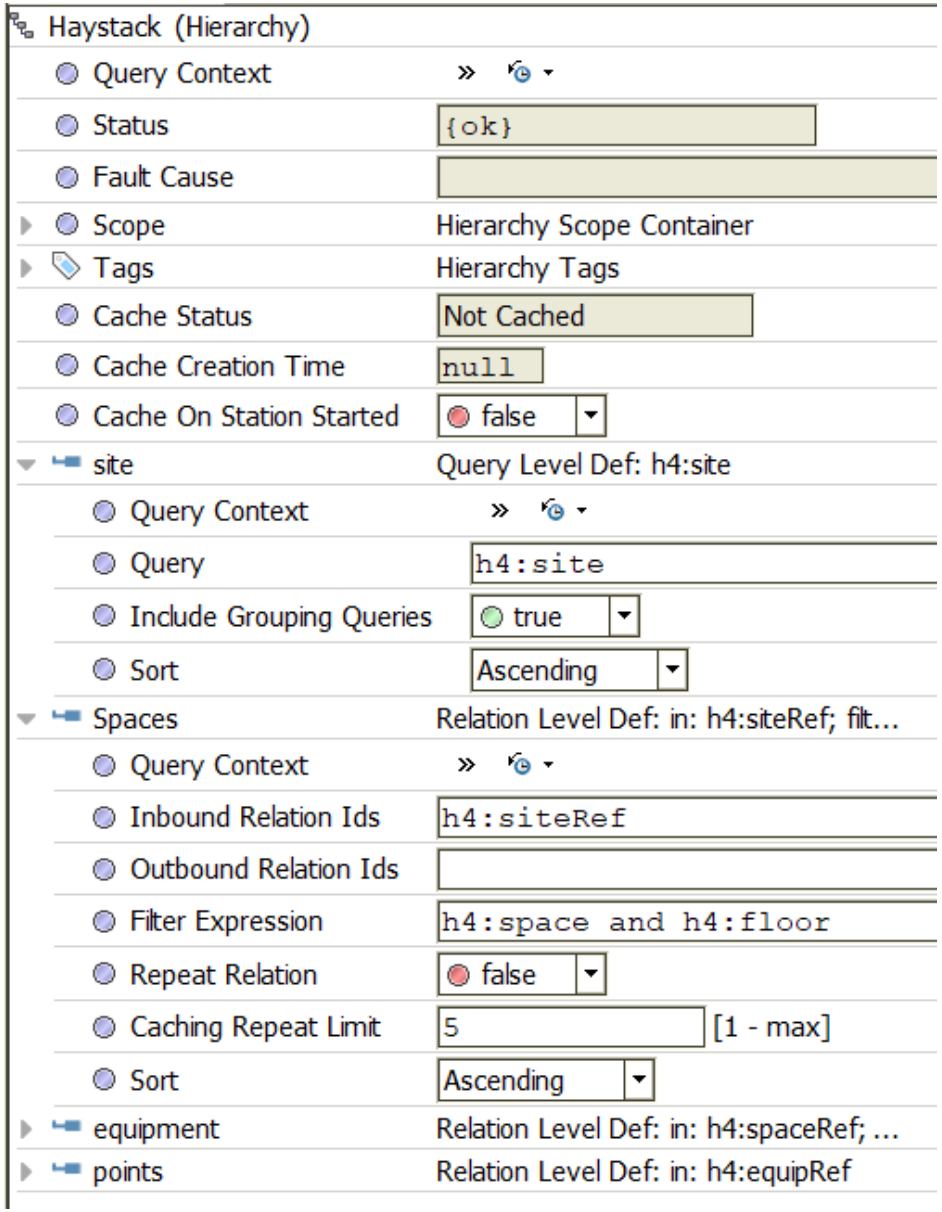
Hierarchies – Niagara Relations



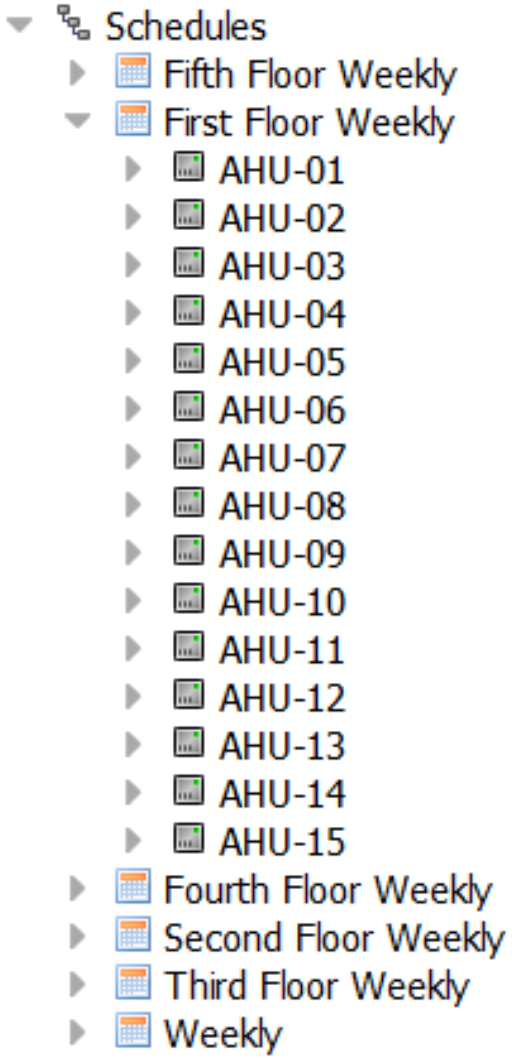


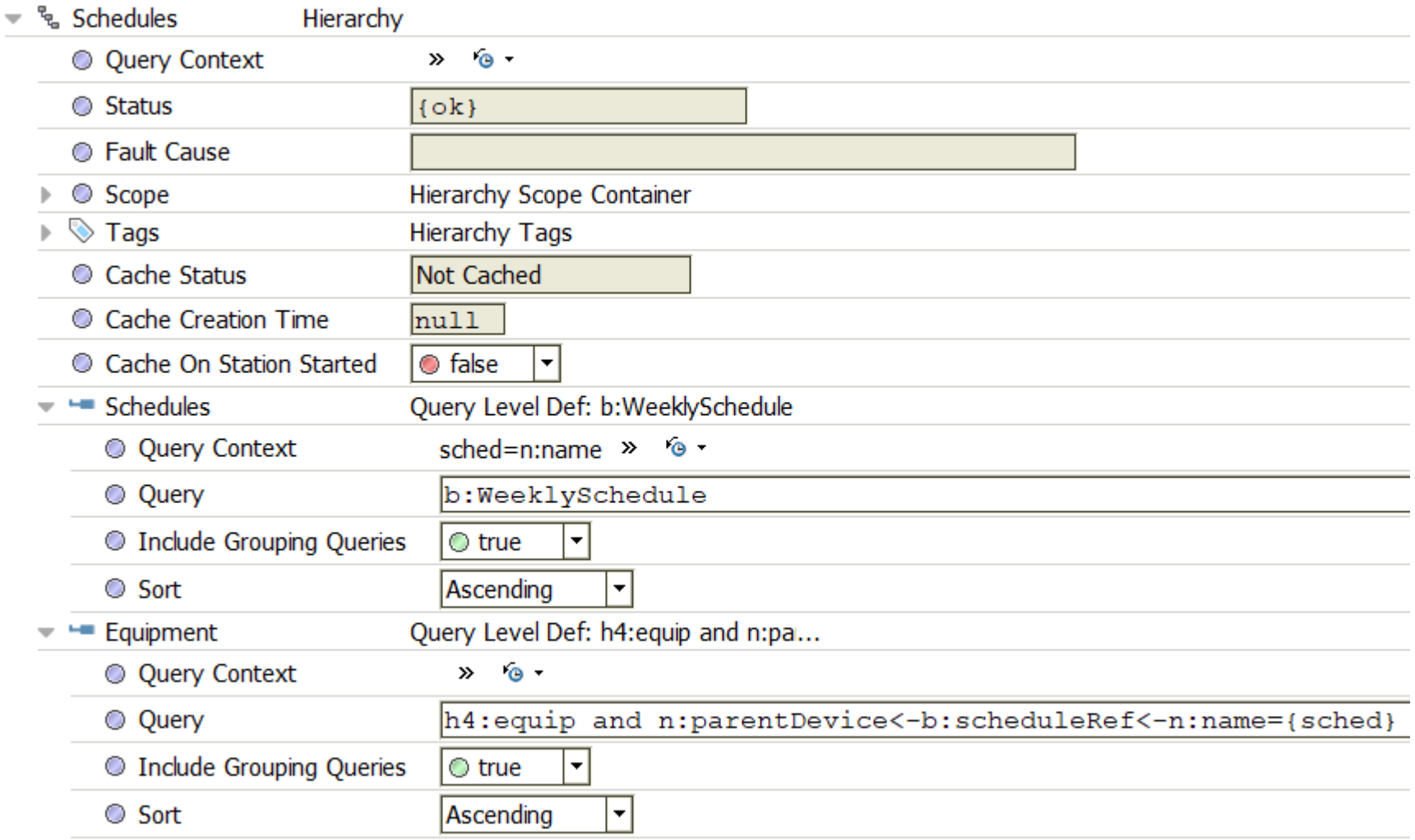
Hierarchies – Haystack Relations



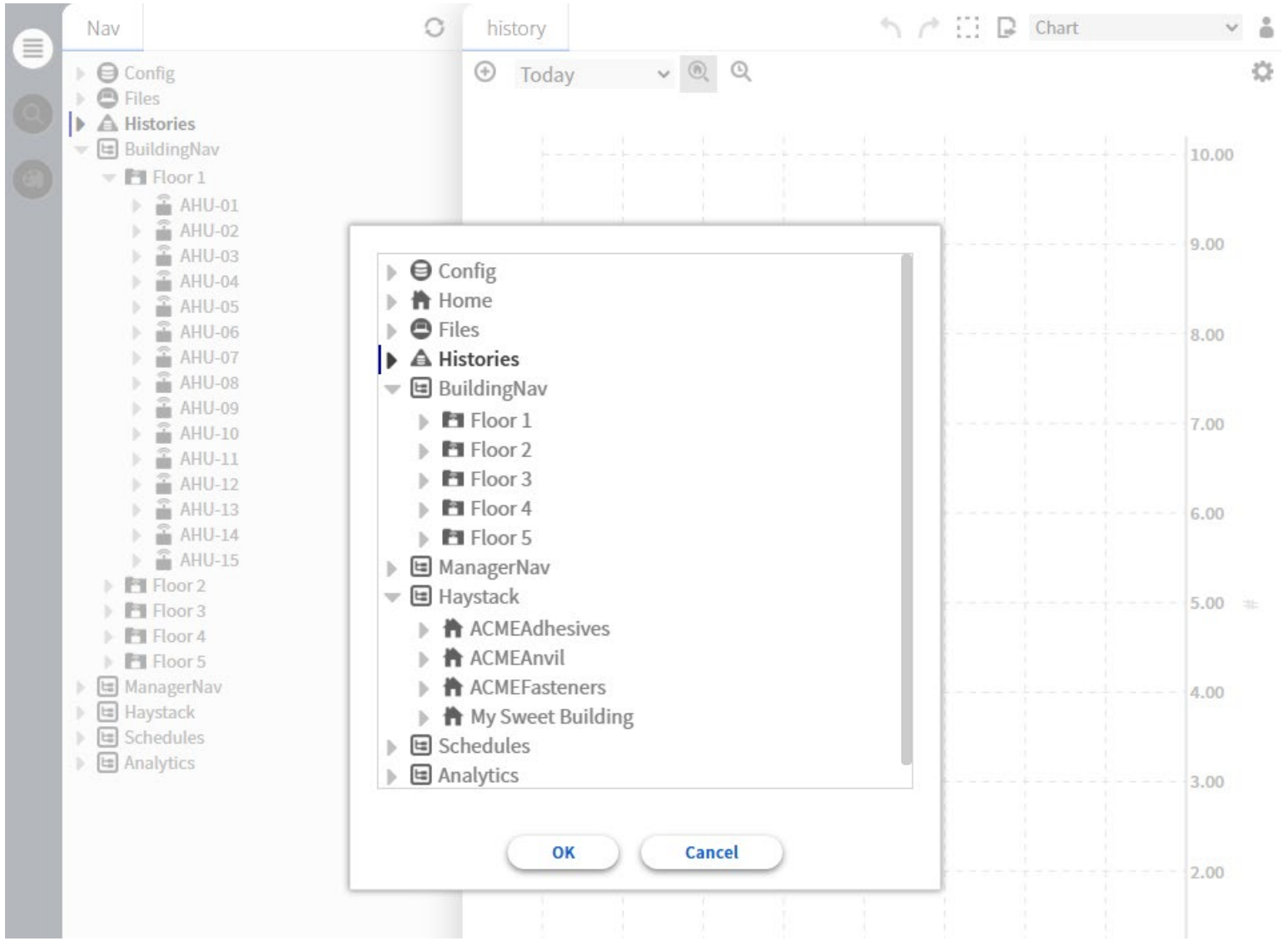


Hierarchies – Schedule Relations





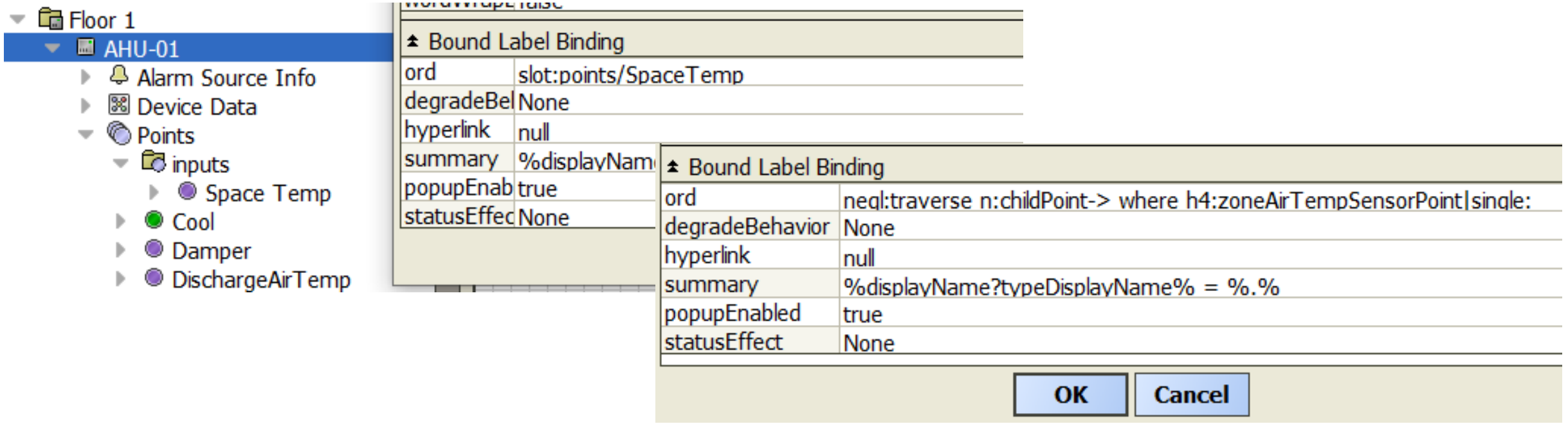
Hierarchies – History Groupings



Graphic Use Cases

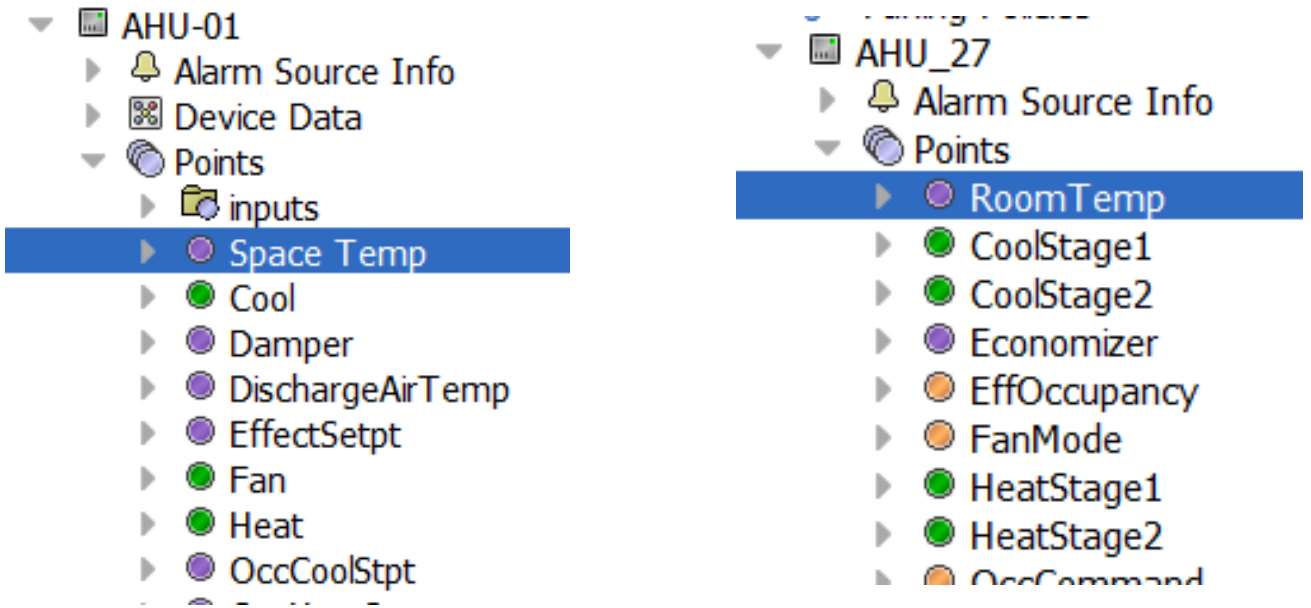
Graphics – Fault Tolerant Design

- Slot path variations between stations.
- Engineering changes such as renaming points, renaming ancestors in the slot path, or moving components.



Graphics – Fault Tolerant Design

- Inconsistent naming conventions.
- Not enforcing project specifications and standards.



Graphics – Related Data

- Using slot path bindings to related data often require unique Px views with absolute ords.

Bound Label Binding

ord	station:|slot:/Drivers/BacnetNetwork/Chiller1/points/CWS Temp
degradeBehavior	None
hyperlink	null
summary	%displayName?typeDisplayName% = %.%
popupEnabled	true
statusEffect	None

Bound Label Binding

ord	neql:traverse h4:chilledWaterRef->|single:|neql:h4:chilledWaterLeavingTempSensorPoint|single:
degradeBehavior	None
hyperlink	null
summary	%displayName?typeDisplayName% = %.%
popupEnabled	true
statusEffect	None

Graphics – Related Data

- Relations support more complex hyperlinks with relative Px files to schedules or other related equipment.

Bound Label Binding	
ord	neql:traverse n:childPoint-> where h4:occupiedSpPoint|single:|neql:traverse b:scheduleRef<-|single:
degradeBehavior	None
hyperlink	neql:traverse n:childPoint-> where h4:occupiedSpPoint|single:|neql:traverse b:scheduleRef<-|single:
summary	%displayName?typeDisplayName% = %.%
popupEnabled	true
statusEffect	None

Analytic Use Cases

Analytics – Rollup & Value Bindings

ACME Corporation

Alarms	Energy	
0	This Month	61486.8 kW-hr
	Last Month	105997.8 kW-hr

ACME Anvil

Power	
Now	349.1 kW
Last Month Max	762.5 kW
Last Month Avg	288.9 kW

Energy	
This Month	28105.5 kW-hr
Last Month	48352.7 kW-hr

Alarms
0

ACME Adhesives

Power	
Now	180.0 kW
Last Month Max	581.0 kW
Last Month Avg	166.4 kW

Energy	
This Month	16232.7 kW-hr
Last Month	28028.9 kW-hr

Alarms
0

ACME Fasteners

Power	
Now	180.0 kW
Last Month Max	525.7 kW
Last Month Avg	176.0 kW

Energy	
This Month	17148.7 kW-hr
Last Month	29616.1 kW-hr

Alarms
0

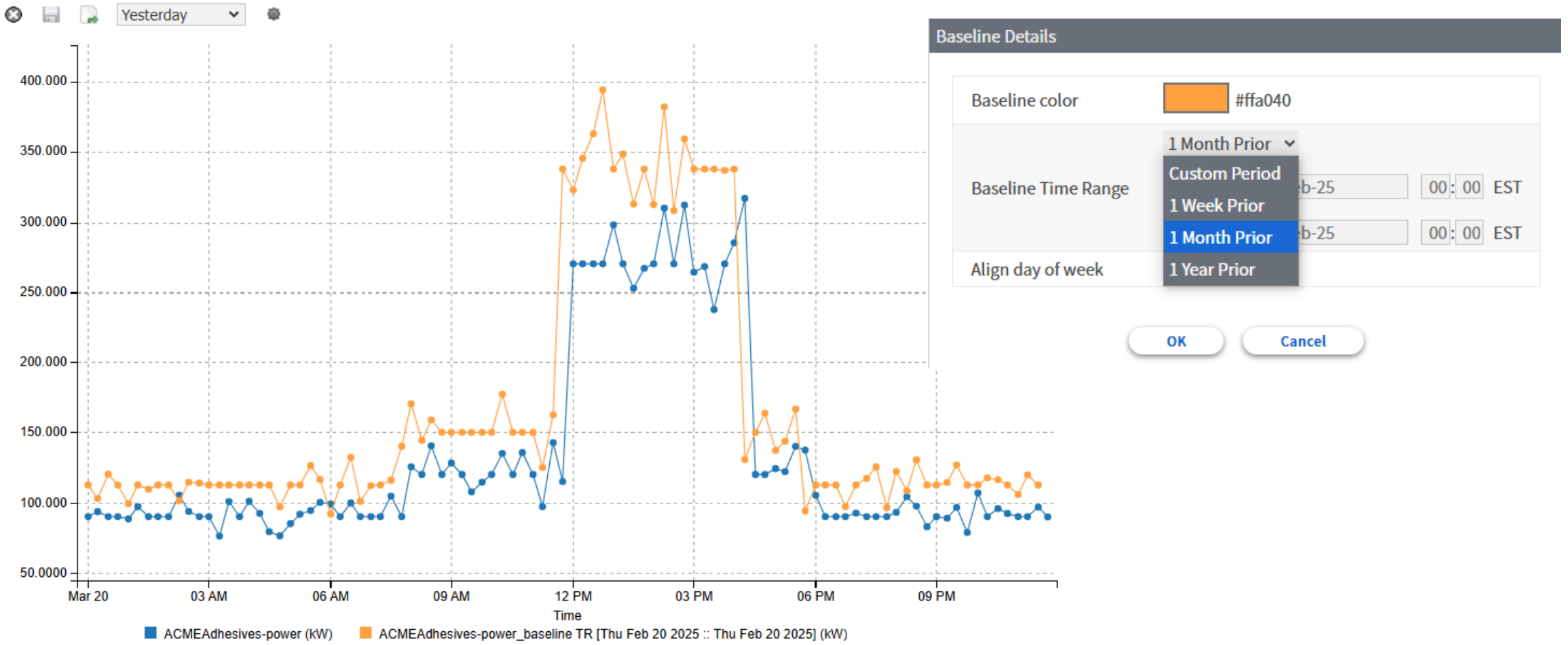
Analytic Rollup Binding

degradeBehavior	None
hyperlink	null
summary	%displayName?typeDisplayName% = %.%
popupEnabled	false
data	h4:energy
node	slot:/ACMECorp
dataFilter	
timeRange	monthToDate
interval	
aggregation	
rollup	
unit	
daysOfWeek	{Sun Mon Tue Wed Thu Fri Sat}
totalize	true
missingDataStrateg	
refreshRate	5 minutes

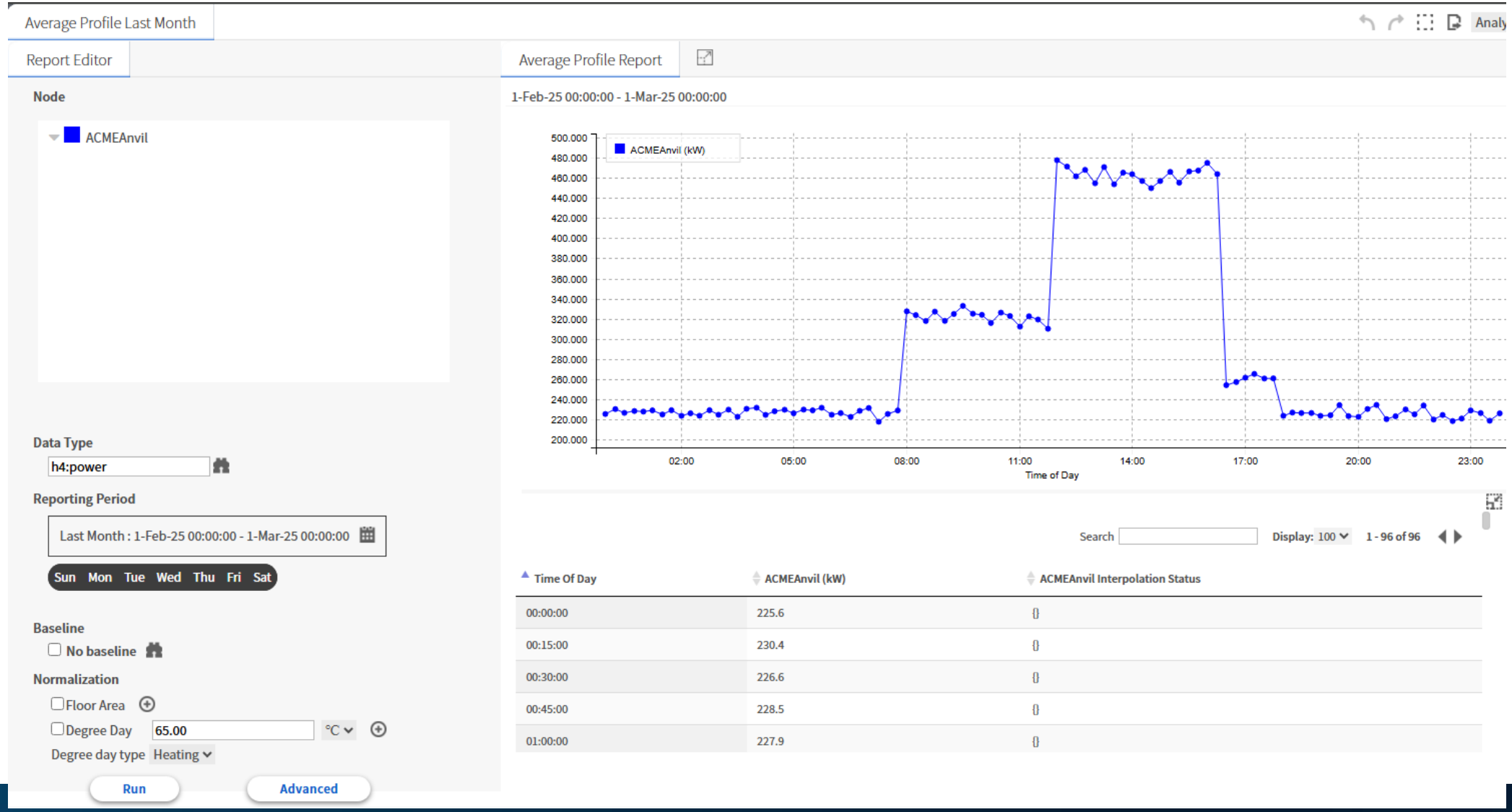
Analytic Value Binding

degradeBehavior	None
hyperlink	null
summary	%displayName?typeDisplayName% = %.%
popupEnabled	false
data	h4:power
node	slot:/ACMECorp/ACMEAnvil
dataFilter	
aggregation	

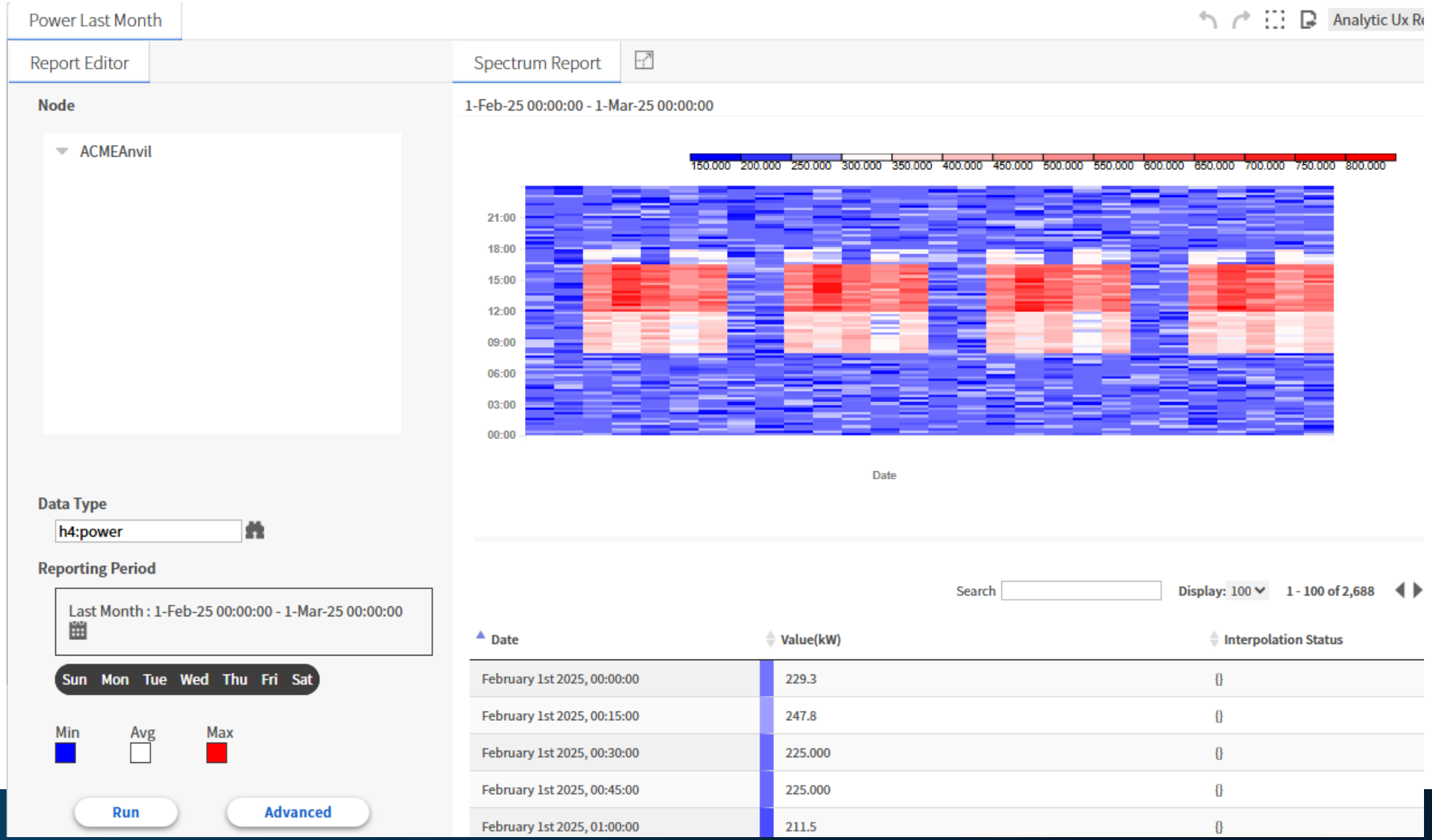
Analytics – Baseline Comparison



Analytics – Charts & Reports

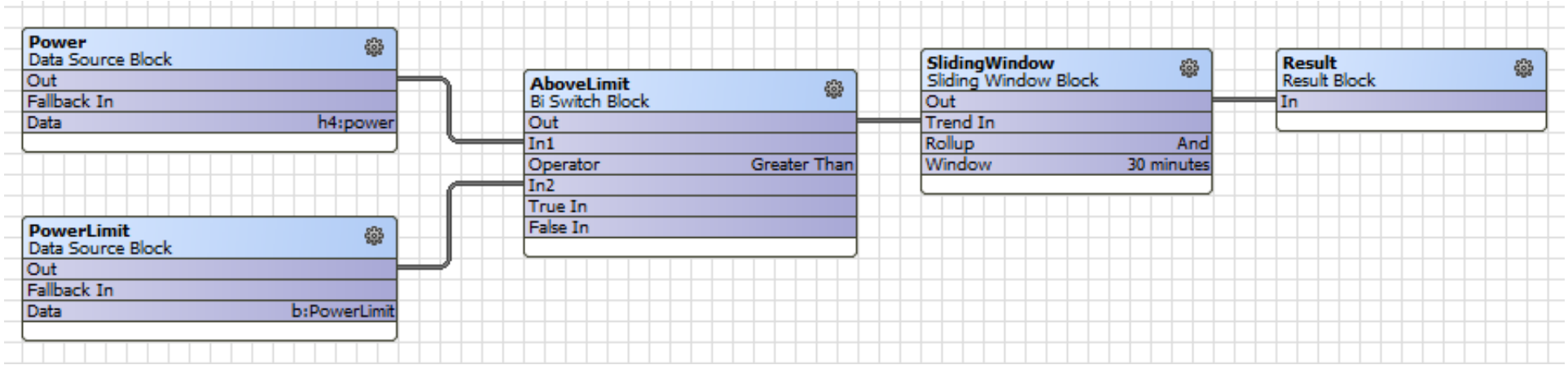


Analytics – Charts & Reports



Analytics – Fault Detection Algorithms

- Single source for fault detection logic.
- Flexibly applied to many components in the station.
- Robust alarming when combined with control points and alarm extensions.



Analytics – Web API

- API requests utilize data model

Modeling with Haystack

NS26

TRIDIUM

Haystack

- Started in 2011
- Project Haystack formed in 2014: project-haystack.org

Haystack 3

- Flexible set of simple tags
- Some point tag groupings
- Site-Equip-Point structure
- Could be applied inconsistently

tags.csv

	A	B
1	name	kind
2	absorption	Marker
3	ac	Marker
4	active	Marker
5	ahu	Marker
6	ahuRef	Ref
7	air	Marker
8	airCooled	Marker
9	angle	Marker
10	apparent	Marker
11	area	Number
12	avg	Marker
13	barometric	Marker
14	blowdown	Marker
15	boiler	Marker
16	bypass	Marker
17	centrifugal	Marker
18	chilled	Marker
19	chilledBeamZone	Marker
20	chilledWaterCool	Marker
21	chilledWaterPlant	Marker
22	chilledWaterPlantRef	Ref
23	chiller	Marker
24	circ	Marker

ahu.txt

```
**Discharge**
discharge air temp sensor
discharge air humidity sensor
discharge air pressure sensor
discharge air flow sensor
discharge air fan cmd
discharge air fan sensor

**Return**
return air temp sensor
return air humidity sensor
return air pressure sensor
return air flow sensor
return air co2 sensor
return air fan cmd
return air damper cmd

**Mixed**
mixed air temp sensor

**Outside**
outside air temp sensor
outside air humidity sensor
outside air pressure sensor
outside air flow sensor
outside air flow sp
outside air damper cmd

**Exhaust**
```

Haystack 3

- Requirements in documentation
  - ahu, equip, hvac
  - rooftop, elecHeat, dxCool
- Not built into the model

19 AHUs

19.1 Overview

The ahu tag is used model air handling equipment designed to heat or cool air. In Project Haystack, packaged rooftop units are considered a special class of AHU. Packaged units use the ahu tag, but should also specify the rooftop tag:

```
ahu                 // any type of air handler unit (built-up or RTU)
ahu and rooftop     // only packaged rooftop units (RTUs)
ahu and not rooftop // only built-up custom AHUs
```

19.2 Tags

AHUs should always be marked as ahu and equip. The following tags are also used:

- hvac: always specified to mark as an HVAC asset
- rooftop: if the AHU is a packaged rooftop unit (RTU)
- mau: if the AHU is a makeup air unit
- chilledWaterPlantRef: reference plant supplying chilled water
- hotWaterPlantRef: reference plant supplying hot water
- steamPlantRef: reference plant supplying steam

19.2.1 Heating and Cooling Method

AHUs should define their heating method using one of the following tags:

- elecHeat
- hotWaterHeat
- steamHeat
- gasHeat

Cooling method should be defined using one of the following tags:

- chilledWaterCool
- dxCool

19.2.2 Constant vs Variable Volume

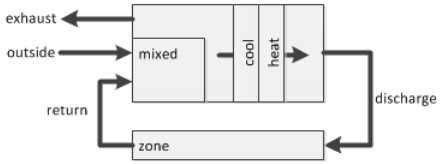
An AHU should be tagged as either constantVolume or variableVolume based on its ability to adjust the volume of air flow. Typically this distinction is based on whether the AHU's fan is single speed or a VFD.

19.3 Sections

Most points in an AHU are associated with one of the following sections of the unit:

- discharge: air exiting the unit to be supplied to the zones/terminal units
- return: air returning from the zone back into the unit
- outside: fresh, outside air entering the unit for air quality and economizing
- exhaust: air exiting the unit back outside
- mixed: return and outside air mixed together before passing through the heating/cooling elements
- cool: cooling elements/coils
- heat: heating elements/coils
- zone: conditioned space associated with the unit

The follow diagram shows the logical flow of air through an AHU:



19.4 Points

The following lists points commonly used with an AHU:

Discharge

- discharge air temp sensor
- discharge air humidity sensor
- discharge air pressure sensor
- discharge air flow sensor
- discharge air fan cmd
- discharge air fan sensor

Return

- return air temp sensor
- return air humidity sensor
- return air pressure sensor

Haystack 3 Dictionary

- Create tags/relations based on a customized tags.csv
- Add columns for smart types (IdTag, UnitTag) and tag rules
- Create tag groups based on equip txt files
- Add smart siteRef and equipRef relations

Haystack 4

- Started around 2019
- Structured set of defs
- Site-Space-Equip-Point structure

Haystack 4

def **ahu**

Air Handling Unit: An enclosure with a fan that delivers air to a space via ductwork and performs one or more of the functions of cleaning, heating, cooling, humidifying, dehumidifying, ventilating or circulating the air. See AHUs chapter.

meta

children	See below
def	ahu
doc	See above
is	airHandlingEquip
lib	lib:phIoT
mandatory	✓
wikipedia	https://en.wikipedia.org/wiki/Air_handler

usage **ahu** **equip**

supertypes

marker	Marker labels a dict with typing information
entity	Top-level dicts with a unique identifier
equip	Equipment asset
airHandlingEquip	HVAC equipment that conditions and delivers air via one or more fans
input	Entity inputs a substance which flows from another entity
elec-input	Entity inputs electricity which flows from another entity
output	Entity outputs a substance with flows to other entities
air-output	Entity outputs air to other entities

subtypes

doas	Dedicated Outside Air System
mau	Makeup Air Unit
rtu	Roof Top Unit

tags

ahuZoneDelivery	AHU delivery method of conditioned air to the zone
airVolumeAdjustability	Ability of air handling equip to adjust volume of air flow
coolingProcess	Processed used to cool a substance
dis	Display name for an entity
ductConfig	Ductwork configuration
ductDeck	Cold, hot, or neutral deck
elecRef	Electricity flows from the referent to this entity
equipRef	Reference to equip which contains this entity
heatingProcess	Processed used to heat a substance
id	Defines the unique identifier of an entity in system using a `ref` value type
siteRef	Reference to site which contains the entity
spaceRef	Reference to space which contains this entity
systemRef	Reference to system

children

- **bypass damper cmd point**
- **cool cmd point**
- **dessicantDehumidifier cmd point**
- **discharge duct equip**
- **economizer duct equip**
- **economizing cmd point**
- **exhaust duct equip**
- **faceBypass cmd point**
- **filter sensor point**
- **freezeStat sensor point**
- **heat cmd point**
- **heatWheel cmd point**
- **humidifier equip**
- **hvacMode sp point**

NS26

TRIDIUM

Haystack 4 Dictionary

- Use standard, unaltered defs.json and protos.json files
- Tags/relations
- Tag rules based on def structure
- Tag groups based on conjuncts and protos
- Extra Niagara json for more tag rules and smart relations
- Can use updated json files as Haystack 4 evolves

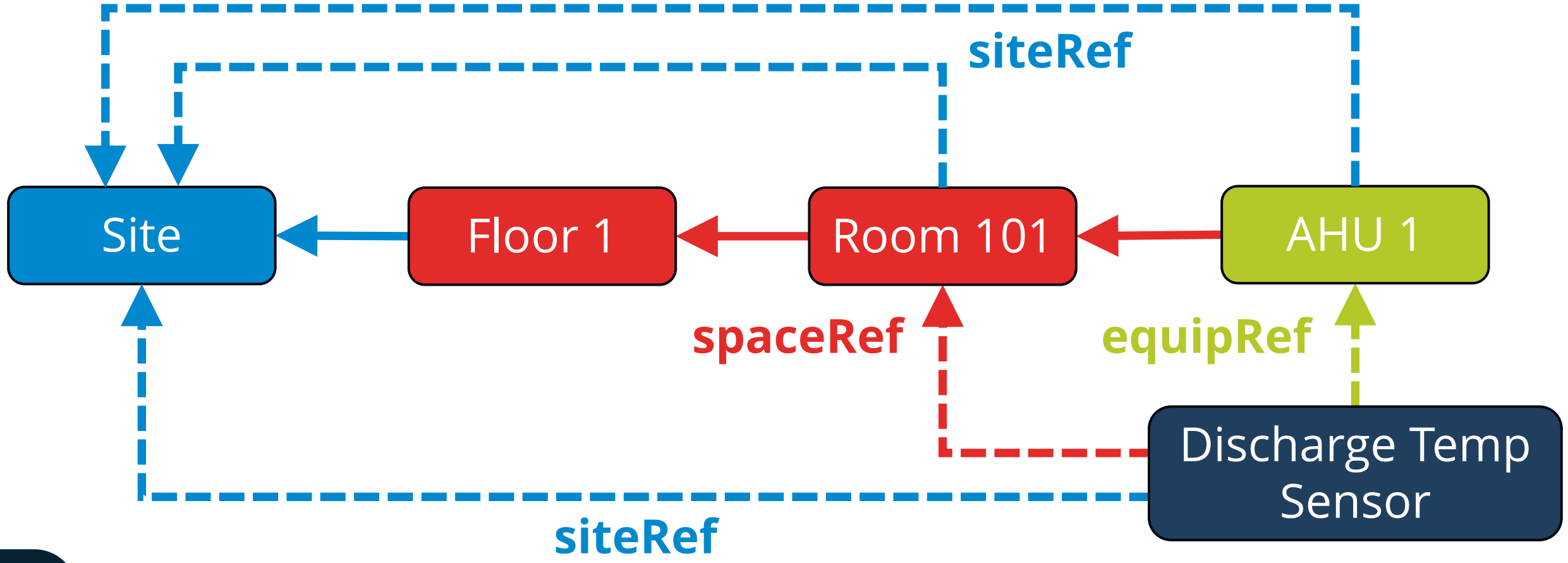
Haystack 4 Dictionary

Showing tags on: AHU_1 ☑ Direct ☑ Implied

Tag Id	Tag Name	Value
h4:ahu	ahu	Marker
h4:dis	dis	AHU_1
h4:equip	equip	Marker
n:device	device	Marker
n:displayName	displayName	AHU_1
h4:device	device	Marker
h4:airHandlingEquip	airHandlingEquip	Marker
h4:entity	entity	Marker
n:type	type	bacnet:BacnetDevice
n:station	station	bacnetTemplate
n:name	name	AHU_1
n:ordInSession	ordInSession	station:|h:6b8df
h4:id	id	nspace:bacnetTemplate~slot:.Drivers.BacnetNetwork.AHU_1

Haystack 4 Smart Relations

Solid line is manual relation, dashed is implied relation



Haystack 5

- Introduced in 2025
- Xeto: github.com/Project-Haystack/xeto
- Community contribs: github.com/Project-Haystack/xeto-cc
- Device templates

Haystack 5

xeto-cc/blob/main/src/xeto/cc.siemens.apogee.tec/specs.xeto

```
// BACnet PTEC Terminal Box (VAV) Controller Application 6520 - VAV Cooling Only
PTEC_6520 : ph::Vav {
  dis: "PTEC 6520"
  applicationNumber: "6520"
  attrs: {
    ElecAcVoltAttr { val: "24" }
    ManufacturerAttr { val: "Siemens"}
  }
  points: {
    ColdDeckDischargeAirFlowSp { bacnetAddr: BacnetAddr { addr: "AO31", dis: "CLG_FLOW_MIN" }, min }
    ColdDeckDischargeAirFlowSp { bacnetAddr: BacnetAddr { addr: "AO32", dis: "CLG_FLOW_MAX" }, max }
    CoolCmd { bacnetAddr: BacnetAddr { addr: "AO79", dis: "CLG_LOOPOUT" } }
    DamperCmd { bacnetAddr: BacnetAddr { addr: "AO48", dis: "DMPR_COMD" } }
    DamperSensor { bacnetAddr: BacnetAddr { addr: "AO49", dis: "DMPR_POS" } }
    DischargeAirFlowSensor { bacnetAddr: BacnetAddr { addr: "AO75", dis: "FLOW" } }
    DischargeAirFlowSp { bacnetAddr: BacnetAddr { addr: "AO76", dis: "CTL_FLOW_MIN" }, min }
    DischargeAirFlowSp { bacnetAddr: BacnetAddr { addr: "AO77", dis: "CTL_FLOW_MAX" }, max }
    DischargeAirFlowSp { bacnetAddr: BacnetAddr { addr: "AO93", dis: "FLOW_STPT" } }
    DischargeAirTempSensor { bacnetAddr: BacnetAddr { addr: "AI15", dis: "SUPPLY_TEMP" } }
    OccupiedSp { bacnetAddr: BacnetAddr { addr: "AO29", dis: "DAY_NGT" } }
```

Haystack 5 Dictionary?

- Should use a standard, unaltered xeto.json
- Should use an updated json file as Haystack 5 evolves
- Probably need extra Niagara info
- Should include community contributions

Haystack 5 Templates?

- Niagara templates based on community contributions
- Drop in, enter device address: tags and proxy exts configured

Modeling with Brick

NS26

TRIDIUM

Brick

- Started in 2016
- brickschema.org
- Merged with RealEstateCore in 1.4.0 (Apr 2024)
- RDF/SHACL
- Classes like Haystack defs

Brick

brick **Air_Handling_Unit**

Definition

Schema

```
class brick Air_Handling_Unit extends brick HVAC_Equipment .
```

Equivalent Classes

- brick AHU
- brick Air_Handler_Unit

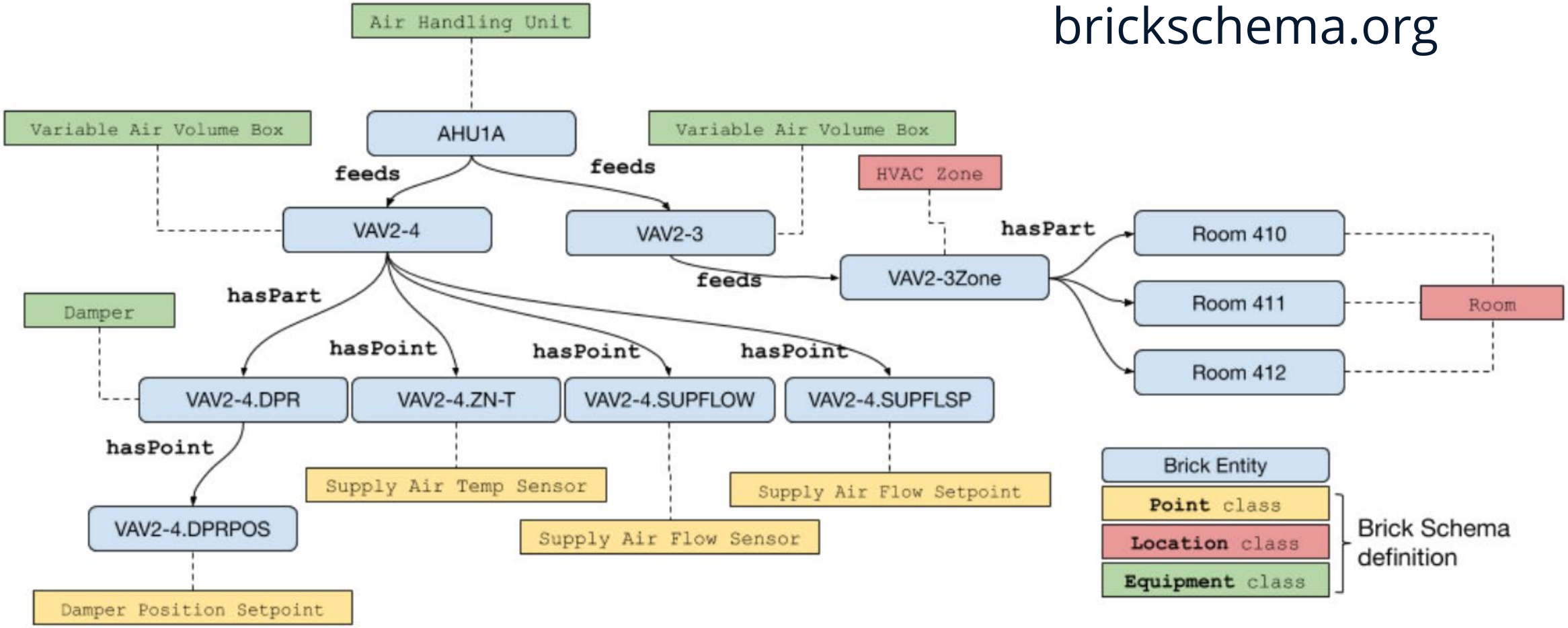
Properties

PROPERTY	VALUE	DESCRIPTION
* brick feeds (inherited from brick HVAC_Equipment)	brick HVAC_Equipment brick Valve rec Space	The subject is upstream of the object in the context of some sequential process; some media is passed between them
* brick hasPart (inherited from brick HVAC_Equipment)	brick HVAC_Equipment brick Valve	The subject is composed in part of the entity given by the object
* brick azimuth (inherited from brick Equipment)	bsh AzimuthShape	(Horizontal) angle between a projected vector and a reference vector (typically a compass bearing). The

- brick hasAssociatedTag tag AHU [from <http://brickschema.org/schema/1.4/Brick>:48147:28]
- brick hasAssociatedTag tag Air [from <https://brickschema.org/schema/1.4/Brick>:48148:9]
- brick hasAssociatedTag tag Equipment [from <https://brickschema.org/schema/1.4/Brick>:48149:9]
- brick hasAssociatedTag tag Handler [from <https://brickschema.org/schema/1.4/Brick>:48150:9]
- brick hasAssociatedTag tag Handling [from <https://brickschema.org/schema/1.4/Brick>:48151:9]
- brick hasAssociatedTag tag Unit [from <http://brickschema.org/schema/1.4/Brick>:48152:9]

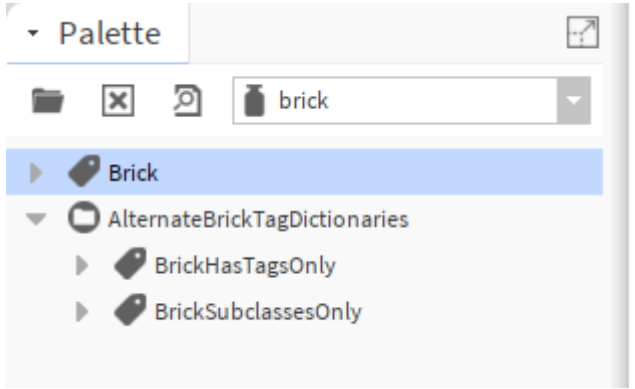
Brick

brickschema.org



Brick Dictionaries

- Use Apache Jena to load brick.ttl
- Library is too large to run on the station
- Generate tag dictionary json files for import
- Brick: class → tag group w/ class tag
- Variations
  - BrickHasTagsOnly: ~Brick + tags
  - BrickSubclassesOnly: different tag group tags



Brick Dictionaries

- No automatic relations: only implied inverse relations
  - For example, add “feeds” relation, get implied “isFedBy” relation
- Does not support SPARQL queries
- Does not support importing/exporting RDF data

ASHRAE 223p

- docs.open223.info
- Lower-level model
- Completed first public review in 2025

Brick ↔ Haystack

- Imply Brick classes based on Haystack defs
- Or, vice versa
- Learn the model you use

Adding the Model

Adding the Data Model

- Niagara template with metadata
- Adding metadata in a template update
- Implied with tag rules
- Added directly with program objects

Questions

Brick and Niagara 4.14+

- Utilizes one smart tag, bk:id, which is applied to points.
- Utilizes one smart relation which implies an inverse relation for specific direct relations.
- Utilizes tag rules to handle some of inference and reasoning from the Brick ontology specification.
- Does not support SPARQL queries.
- Does not support importing/exporting RDF data.

Brick Inverse Relationships

In all cases where we have subject relationship object, an equivalent statement is object inverse-relationship subject

Relationship	Inverse Relationship
hasPoint	isPointOf
hasPart	isPartOf
hasLocation	isLocationOf
feeds	isFedBy
hasSubMeter	isSubMeterOf
hasAssociatedTag	isAssociatedWith
hasTag	isTagOf
meters	isMeteredBy

Core Enhancements

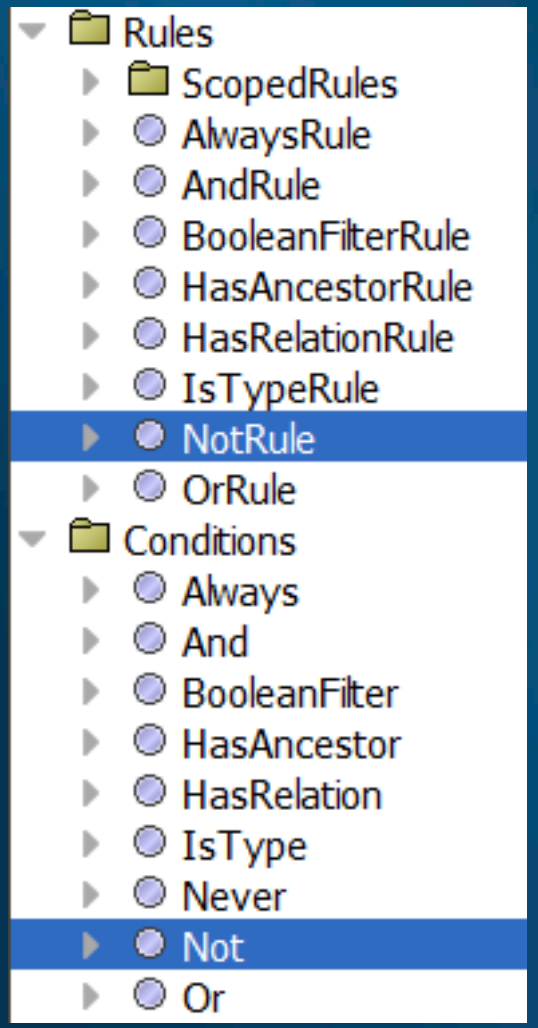
NS26

TRIDIUM

Niagara 4.14

- Additional flexibility to create tag rules using Not rules and conditions.
- New rule implies n:alarmPoint marker tag when a control point has an alarm extension.

n:point	Marker
n:alarmablePoint	Marker
n:input	Marker



Niagara 4.15 – Relation Manager