Jefferson Lab

# What does AI-Ready Accelerator Data mean?

*Nuclear AI-Ready Accelerator Data (NARAD) — an operational-layer ontology and data standard for DOE accelerator facilities*

## FACILITY ONTOLOGY

**Controls, lattice, diagnostics, signals, and operational context**

- Element & signal semantics
- Ontological channel-finding
- Device → process → system hierarchy
- Support live/archived data: state, setpoints, readbacks
- Provenance, units, uncertainty
- Standards-based

### 01 · AGENTIC WORKFLOWS

**Ontological input for agents (e.g. Osprey)**

Databases/knowledge graphs that LLM agents can query to reason about device relationships, dependencies, and operational intent

### 02 · DIGITAL TWINS and CO-DESIGN

**Shared schema for digital twin frameworks**

Common element, signal, and state vocabulary so twins can be populated, synchronized, and compared across facilities

### 03 · AMERICAN SCIENCE CLOUD

**Export of facility state/context for publication**

Structured, governed, FAIR publication of accelerator state and metadata so cross-facility AI can consume it at scale

*Collaboration: JLab, BNL, LBNL, PNNL, and Cornell – funded FY26/FY27 through AmSC Data Provider Partnerships*

*Motivated substantially by Tennant arxiv [“A Core Ontology for Particle Accelerators: Interoperable Data and Workflows...”]*

# CEBAF Application (MaLAPA'26)

*An authoritative facility database (CED) and an emerging community standard (PALS). Each solves half the problem.*

## CED

CEBAF Element Database

FACILITY-SPECIFIC | IN PRODUCTION | AUTHORITATIVE

### What it holds today

- Element inventory: magnets, cavities, BPMs, correctors, diagnostics
- Position, length, s-coordinate along the machine
- EPICS PV names bound to many controllable devices
- Calibration and survey data with provenance
- Region and beamline groupings used by operations

***Gaps:*** *no standard schema — CEBAF-shaped, with CEBAF-specific tables and conventions.*

CED IPAC'15 paper link

## PALS

Particle Accelerator Lattice Standard

FACILITY-AGNOSTIC | EMERGING STANDARD | LATTICE-FIRST

### What it defines today

- Common element types, parameters, and lattice topology
    - Basis for a *lattice* ontology but not controls ontology
- Cross-code interchange (MAD-X, elegant, Bmad, ...)
- Designed for cross-facility reuse and tooling
- Active collaboration across DOE labs and universities
- Extension points for facility-specific semantics

***Gaps:*** *no operational layer — no control channels, no live state, no EPICS bindings*

https://pals-project.readthedocs.io/en/latest/

# CED + Extended PALS → NARAD for CEBAF Knowledge Graph

*The integration pipeline: facility truth on the left, standards-based KG on the right, custom extensions in the middle.*

SOURCE

## CED

*CEBAF Element Database*

- Device inventory & geometry
- Some EPICS PV bindings
- Region / beamline tables
- Calibration, survey, provenance

*Facility truth, as it lives today*

BRIDGE

## PALS + Extensions

*Standards + ontology add-ons*

- PALS core lattice schema
- EPICS → element mapping
- Region / signal-type ontology
- Unit, provenance, uncertainty

*Reusable semantic/ontology layer*

PRODUCT

## Knowledge Graph

*Devices, signals, regions as typed nodes*

- Typed nodes & typed edges
- Cypher-queryable
- Live state bound to devices
- Machine-interpretable context

*What agents actually consume*

CED provides **facility-specific content**. PALS provides the **shared grammar**. Custom extensions map **EPICS channels → lattice elements**, yielding a NARAD KG that is both authoritative and standards-compliant.

Jefferson Lab

Evolving Architecture

# Three-layer PALS-Ext architecture (MaLAPA'26)

*Separates facility-independent physics semantics from facility-specific controls implementation.*

*Facility Agnostic*

CAPABILITY LAYER

1

## What a device can do or report

- Quantities: cavityGradient, magnetCurrent, beamLossRate, …
- Roles: setpoint, readback, status, limit · Units + constraint types
- No facility naming, no PVs — just physics affordances of a device class

*Facility Agnostic*

INTEROPERABLE SEMANTIC SIGNAL LAYER

2

## Interoperable interfaces for capabilities

- Each signal = quantity + role + units + implements → capability
- The unit of interoperability for tools, workflows, and agents
- "All gradient-setpoint signals upstream of detector X" — no PV needed

*Facility-Specific*

FACILITY-SPECIFIC CONTROLS BINDING LAYER

3

## Signals → concrete control-system channels

- EPICS PVs, archiver keys, access protocols (CA, PVAccess, REST, ADO)
- Permissions, cadence, operational metadata
- Multiple bindings per signal: live / archive / simulation

PALS-Ext · RF cavity gradient · YAML

```yaml
RFCavityGradientProfile:
  kind: PALSExtProfile
  PALSExtP:
    capabilities:
      - { id: cap:gradient_setpoint,
          kind: setpoint,
          quantity: cavityGradient,
          units: MV/m }
      - { id: cap:gradient_readback, … }
      - { id: cap:gradient_max_hard, … }
    signals:
      - { id: sig:gradient:setpoint,
          role: setpoint, units: MV/m,
          implements: cap:gradient_setpoint }
      - { id: sig:gradient:readback, … }

R221:    # cavity 1 of CM22
  kind: RFCavity
  PALSExtP:
    uses_profiles: [RFCavityGradientProfile]
    bindings:
      facility: CEBAF
      control_system: EPICS
      channels:
        - { signal: sig:gradient:setpoint,
            pv: "R221:GSET", direction: write }
        - { signal: sig:gradient:readback,
            pv: "R221:GMES", direction: read }
```

Jefferson Lab

# Status and next steps (MaLAPA'26)

*Where we are today, what's coming, and how this feeds back into the NARAD framing.*

TODAY

## CED + PALS subset, in hand

A CEBAF region lifted into PALS form, wired to EPICS through a custom ontology extension, exposed as a Cypher-queryable KG.

NEXT

## LinkML schema development

- Develop a larger LinkML schema for verification and validation
- Develop a framework for standards-based ontology and knowledge graphs

THEN

## Cross-facility uptake

Implement the schema and patterns to partner facilities (BNL, LBNL, other volunteers) so NARAD evolves as a multi-facility standard

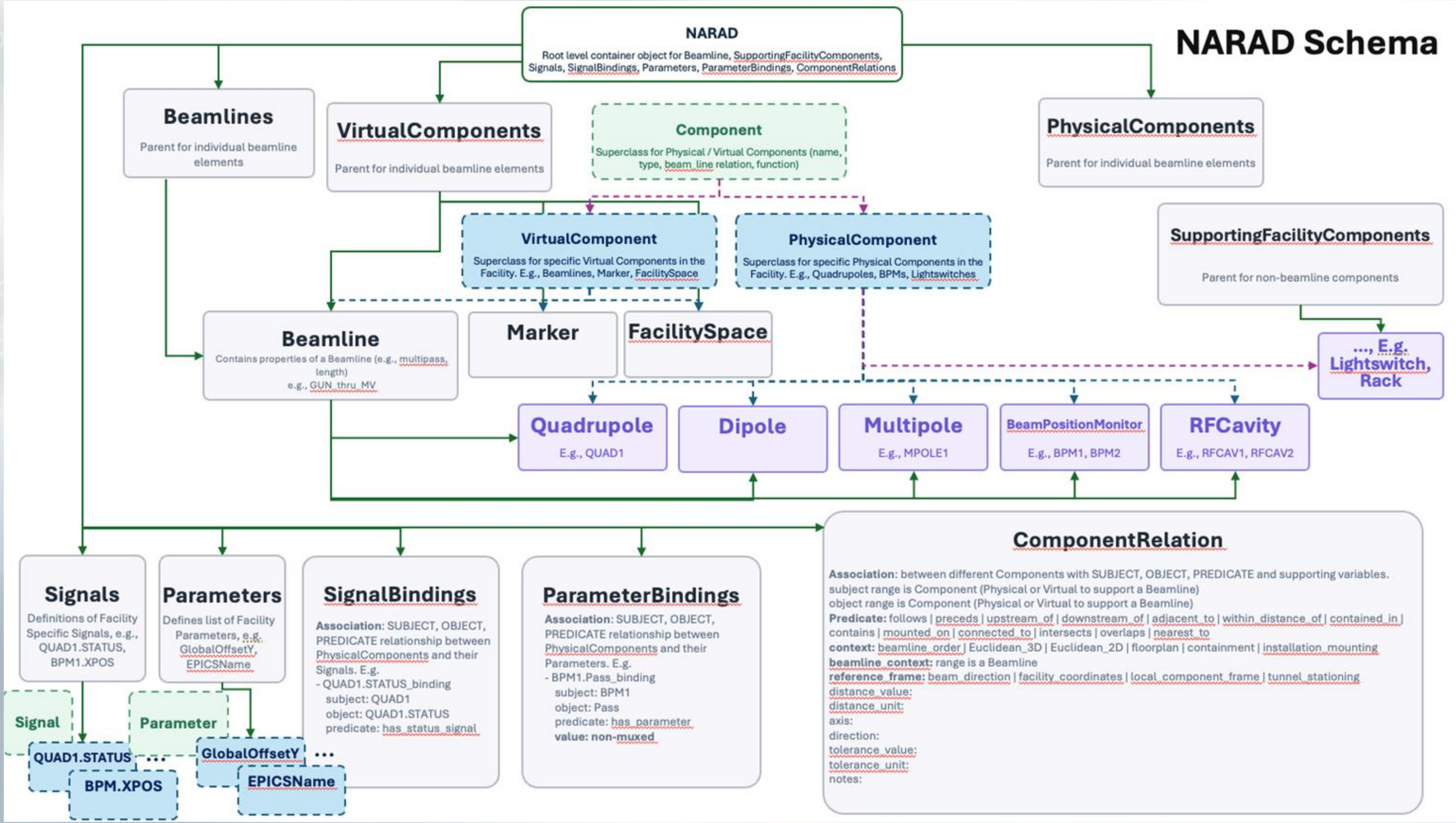
WHERE THIS PLUGS IN · back to the NARAD framing

01 **Agentic workflows**
Osprey and other LLM-driven optimizers query the KG for authoritative facility context.

02 **Digital twins**
Sample digital twins share the same vocabulary and ground truth so they can be populated and compared.

03 **American Science Cloud**
FAIR publication including via MOAT — facility state and context, consumable by cross-facility AI.

# NARAD Schema Evolution (Diana McSpadden, JLab; Jun 29 2026)




- Facility-independent base schema (“design, conventions, and standards”)
- Facility-specific schema extensions (“implementation”)

- Subject/Predicate/Object
- triples for knowledge
- graph ontology embedding

- Balance
  - Extensible: descriptive over proscriptive
  - Standards-based for ontological portability

# Draft paper for peer-reviewed submission

- 19pp at present, invited to special edition journal
- Motivation, approach, innovations
  - Target Use Cases
    - facility AI-ready descriptions/channel finding
    - facility context for published/exported accelerator data
    - AI-ready facility ground truth for twins, co-design
  - Design Principles: LinkML, layered ontology
  - Example use case: JLab UITF facility

```
signal_layer:
  facilities:
    UITF:
      facility_identifier:
        - {identifier_type: doi, identifier_value:
     10.1016/j.nima.2022.167093}
      control_system: {name: epics, naming_convention: <
     device><signal_code>}
      signal_definitions:
        magnet_signals:
          device_type: Magnet
          signal_bindings:
            current_setpoint: {control_system_suffix: .S}
            current_readback: {control_system_suffix: M}
            magnet_status:    {control_system_suffix: .
     STAT}
        quadrupole_signals:
          device_type: Quadrupole
          signal_bindings:
            strength_setpoint: {control_system_suffix: .
     BDL}
```

```
- name: MQWK202
  length: 0.125
  is_instance_of: Quadrupole
  instance_signal_bindings:
    current_setpoint:
      pv_name: MQWK202.S
      is_instance_of: QuadrupoleCapability.signals.
     current_setpoint
      for_property: current_setpoint
      is_hosted_by: UITF
      is_property_of: epics
    strength_setpoint:
      pv_name: MQWK202.BDL
      is_instance_of: QuadrupoleCapability.signals.
     strength_setpoint
      for_property: strength_setpoint
      is_hosted_by: UITF
      is_property_of: epics
  param_set:
    Kn1: 0.0
    facility_param_set:
      facility_element_type: QW        # facility-local
     type name
      modeled_as: KQUAD                # simulation-code
     element
      properties:
        Controlled_by: iocitfmag
        ControlsUnits: Amps
        EPICSName: MQWK202
        # ... survey, Twiss, and hysteresis fields ...
```

## NARAD: A LinkML-Based Layered Ontology for AI-Ready Particle Accelerator Facility Data

The NARAD Collaboration[1*]

[1*]OrgDiv, OrgName, OrgAddress, OrgCity, OrgState, xxxxx, USA.

Corresponding author(s). E-mail(s): authorEmail@org;

**Abstract**

We present NARAD (Nuclear AI-Ready Accelerator Data), a LinkML-based operational ontology and knowledge-graph framework for particle accelerator facilities. Realizing AI-assisted particle accelerator operations requires more than access to control-system channels: agents must know what a channel represents, which device owns it, what units it uses, and how it relates to machine models. NARAD separates accelerator facility context into layers: beamline layout, devices, capabilities, signals, and facility-specific bindings. This separation allows stable accelerator concepts, such as quadrupoles and beam-position readbacks, to be modeled independently from local control channel names. NARAD implementation defines a schema for accelerator operational data, records mappings to standards including PALS for accelerator lattices, SOSA/SSN for signals, and QUDT for units, and generates validated instance data and graph artifacts for downstream use and interrogation. We demonstrate the model with Jefferson Lab Upgraded Injector Test Facility (UITF) and Continuous Electron Beam Accelerator Facility (CEBAF) injector facilities, and the Brookhaven Booster-to-AGS transport line, including explicit signal-to-control-point bindings and model-generation metadata. The examples show how a query or agent can traverse from an accelerator concepts to concrete control endpoints while preserving provenance. We discuss how NARAD complements lattice-centered standards such as the Particle Accelerator Language Standard (PALS) and Lattice Architecture for a Unified Representation of Accelerators (LAURA), and identify remaining work necessary for cross-facility vocabulary agreement, broader device coverage, and quantitative query benchmarks.

**Keywords:** particle accelerators, ontology, knowledge graph, control systems, LinkML, FAIR data, machine learning