Framework progress — developers' perspective

DUNE framework taskforce and developer workshop
19 September 2024

Workshop discussion format

The developers will share (via Zoom) discussion points for each topic.

Discussion is intended to be unstructured and organic; we'll go where we need to.
- Aim for high-level discussions except when the details become necessary.

Both on-site and remote participants should raise their hands and wait to speak until acknowledged by the moderator.

On-site participants should refrain from "side conversations".

All participants are encouraged to contribute to the live notes.

🔷 Fermilab

Framework requirements evaluation

- DUNE and the developers have been evaluating and iterating about requirements and discussing ideas since February.

- The first iteration of evaluating the requirements is complete.

- Based on agreement from DUNE in June, the developers have become custodians of the requirements and adopted Jama Connect as the requirements-management tool.

🔷 Fermilab

Framework requirements evaluation

- DUNE and the developers have been evaluating and iterating about requirements and discussing ideas since February.

- The first iteration of evaluating the requirements is complete.

- Based on agreement from DUNE in June, the developers have become custodians of the requirements and adopted Jama Connect as the requirements-management tool.

- **Regular involvement from DUNE now necessary for making progress**

    41 requirements ready for DUNE's approval

- We hope to make significant progress on many of the remaining requirements during this workshop.

☐ DUNE-DUNE_STKH-1	Ready for Approval	Algorithm Decomposability
☐ DUNE-DUNE_STKH-2	Ready for Approval	Data Product Representation
☐ DUNE-DUNE_STKH-3	Ready for Approval	Modularity of Services
☐ DUNE-DUNE_STKH-7	Ready for Approval	Multiple persistent data formats
☐ DUNE-DUNE_STKH-8	Ready for Approval	Full utilization of DUNE computing resources
☐ DUNE-DUNE_STKH-9	Ready for Approval	Algorithm hardware requirements
☐ DUNE-DUNE_STKH-10	Ready for Approval	Algorithms can use multiple CPUs
☐ DUNE-DUNE_STKH-11	Ready for Approval	Algorithms can use a GPU

🔷 Fermilab

Framework requirements evaluation

- DUNE and the developers have been evaluating and iterating about requirements and discussing ideas since February.

- The first iteration of evaluating the requirements is complete.

- Based on agreement from DUNE in June, the developers have become custodians of the requirements and adopted Jama Connect as the requirements-management tool.

- **Regular involvement from DUNE now necessary for making progress**

  41 requirements ready for DUNE's approval

- We hope to make significant progress on many of the remaining requirements during this workshop.

☐ DUNE-DUNE_STKH-1	Ready for Approval	Algorithm Decomposability
☐ DUNE-DUNE_STKH-2	Ready for Approval	Data Product Representation
☐ DUNE-DUNE_STKH-3	Ready for Approval	Modularity of Services
☐ DUNE-DUNE_STKH-7	Ready for Approval	Multiple persistent data formats
☐ DUNE-DUNE_STKH-8	Ready for Approval	Full utilization of DUNE computing resources
☐ DUNE-DUNE_STKH-9	Ready for Approval	Algorithm hardware requirements
☐ DUNE-DUNE_STKH-10	Ready for Approval	Algorithms can use multiple CPUs
☐ DUNE-DUNE_STKH-11	Ready for Approval	Algorithms can use a GPU

The goal is to have a complete enough description of framework features to make a plan in anticipation of LBNC meeting in October.

🛠 **Fermilab**

Use cases and user stories

The physics use cases have guided the statements of the requirements.

We would like a record of them, collected in one place, and associated with the stated requirements.

🏛️ **Fermilab**

Use cases and user stories

The physics use cases have guided the statements of the requirements.

We would like a record of them, collected in one place, and associated with the stated requirements.

Fortunately, Jama lets you do this through user stories. The general template:

As a [user role], I can [activity], so that [business value].

The developers have added user stories for the examples found in the original requirements document.

(e.g.)

As a	reconstruction job-runner
I can	overlay simulated and real data
So that	we can approximate the detector's response to physics interactions.

🔳 **Fermilab**

Use cases and user stories

The physics use cases have guided the statements of the requirements.

We would like a record of them, collected in one place, and associated with the stated requirements.

Fortunately, Jama lets you do this through user stories. The general template:

As a [user role], I can [activity], so that [business value].

The developers have added user stories for the examples found in the original requirements document.

(e.g.)

As a	reconstruction job-runner
I can	overlay simulated and real data
So that	we can approximate the detector's response to physics interactions.

We would like to add more user stories throughout this workshop. Please add them at:

https://docs.google.com/spreadsheets/d/1TBCQIwfWYNkM0LXHX-IPP3cGxDImKfSuXlpIC36OU98/edit?usp=sharing

🔀 **Fermilab**

Prioritized discussion topics

Before starting development in earnest, the developers need to understand DUNE's requirements regarding:

1. Flexible processing units
2. Nature of data products
3. Persistency of data products
4. Checkpointing, failures and errors
5. Backwards compatibility wrt. existing art/ROOT files and existing algorithms

In good enough shape

🎇 Fermilab

Prioritized discussion topics

Before starting development in earnest, the developers need to understand DUNE's requirements regarding:

1. Flexible processing units
2. Nature of data products
3. Persistency of data products
4. Checkpointing, failures and errors
5. Backwards compatibility wrt. existing art/ROOT files and existing algorithms

In good enough shape

6. Resource requirements and monitoring
7. Supernova physics
8. Time-ordering and data adjacency
9. Configurability and configuration language

Workshop focus

🎶 Fermilab

Prioritized discussion topics

Before starting development in earnest, the developers need to understand DUNE's requirements regarding:

1. Flexible processing units
2. Nature of data products
3. Persistency of data products
4. Checkpointing, failures and errors
5. Backwards compatibility wrt. existing art/ROOT files and existing algorithms

In good enough shape

6. Resource requirements and monitoring
7. Supernova physics
8. Time-ordering and data adjacency
9. Configurability and configuration language

Workshop focus

10. Random number generation

If there's time

🎔 **Fermilab**

Prioritized discussion topics

Before starting development in earnest, the developers need to understand DUNE's requirements regarding:

1. Flexible processing units
2. Nature of data products
3. Persistency of data products
4. Checkpointing, failures and errors
5. Backwards compatibility wrt. existing art/ROOT files and existing algorithms

In good enough shape

6. Resource requirements and monitoring
7. Supernova physics
8. Time-ordering and data adjacency
9. Configurability and configuration language

Workshop focus

10. Random number generation

If there's time

Strong preference: finish discussing one topic before starting another.

🔶 **Fermilab**

Proposal for how to create and use milestones

- Deliverable milestones specify the set of required functionalities that are met

  - Each milestone must be associated with a requirements baseline

  - Each milestone is verified by successful execution of tests intended to verify the requirements

- Near future deliverable milestones are clearer and more detailed than those in the far future

  - As time marches forward, later milestones are continually refined

  - This gives DUNE the flexibility to control the design and development process based on continually improving information and project needs.

- Calendar milestones cannot go into many specifics.

  - These are most likely what LBNC wants.

Fermilab