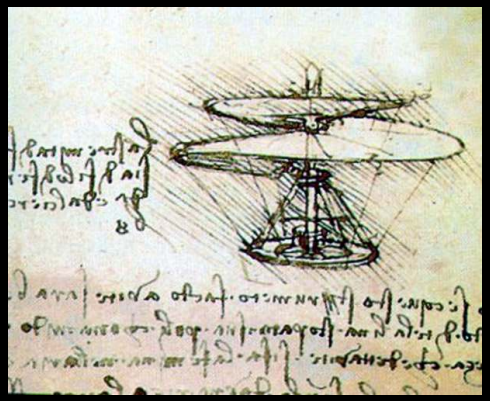

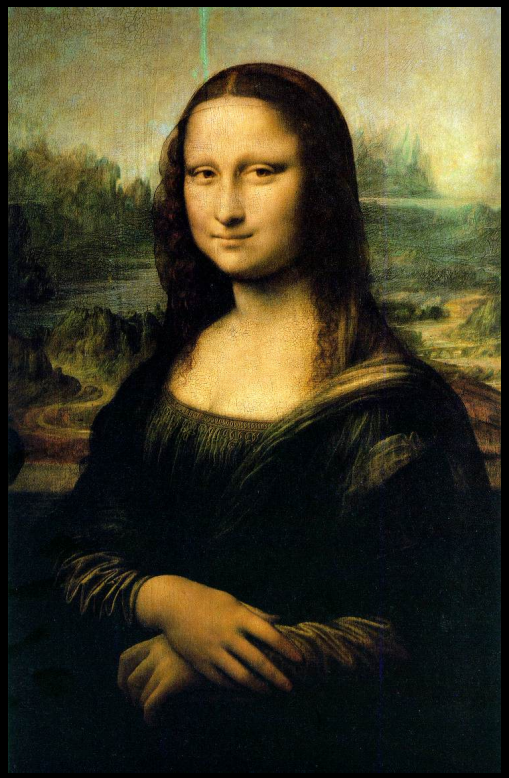

Creating an Assns

Chris Green.

art/ **LArSoft** course.

August 6, 2015.

Fermilab

Fermi National Accelerator Laboratory

Office of Science / U.S. Department of Energy

Managed by Fermi Research Alliance, LLC

What is an Assns (Redux)?

What is an Assns?

- An `art::Assns` is a data product representing *bidirectional associations* between items in collections (in an `art::Event`) of objects of *different types*.
- In addition to recording the fact of an association between such items, it can also save an object recording information specific to the association, such as with `art::Assns<Track, Hit, ResidualInfo>`.

What is an Assns, really?

- An `art::Assns<A, B[, D]>` is a wrapper around one or two sequences: the first being `std::vector<std::pair<art::Ptr<A>, art::Ptr<B>>>` and the second being `std::vector<D>` if appropriate.

Constructing and filling an Assns

Constructing and filling an art::Assns

- Assuming `e` is a (reference to) `art::Event`:

```
// Construct on the heap.
auto coll =
std::make_unique<art::Assns<A, B[, D]>>();
// Fill ...
// Put into the event.
e.put(coll);
```

- `void art::Assns<A, B>::`
  `addSingle(Ptr<A> const &,`
  `Ptr<B> const &);`

- `void art::Assns<A, B, D>::`
  `addSingle(Ptr<A> const &,`
  `Ptr<B> const &,`
  `D const &);`

Where do the Ptrs come from?

- ```
  art::Handle<TrackCollection> h;
  e.getByLabel(label, h);
  art::Ptr<Track> p(h, 2);
  ```

- ```
  auto coll = make_unique<TrackCollection>();
  // ...
  auto pid = e.put(std::move(coll));
  art::Ptr<Track>
    p(pid, 2, e.productGetter(pid));
  ```

Using an Assns Directly

Using an Assns Directly

```
art::Handle<art::Assns<A, B, D>> h;
e.getByLabel(label, h);
auto & coll = *h;
for (auto i = coll.begin(), ei = coll.end();
     i != ei;
     ++i) {
  A const & a = *i->first;
  B const & b = *i->second;
  D const & d = coll.data(i);
  // ...
}
```

Noteworthy Nuggets

Noteworthy Nuggets

- The order of association on retrieval does not depend on the order of association on production: an `art::Assns<A, B, D>` as stored in the event may be retrieved alternatively (or additionally) as an `art::Assns<B, A, D>`. This carries over to the *SQOs*.
- A given *Assns* may contain associations between multiple collections of `A` and / or `B` without effort from the creator or the user of the *Assns* beyond constructing the `art::Ptrs` correctly for the `addSingle()` calls.