art - Feature #11026

Tuple product support

11/30/2015 02:46 PM - Jim Kowalkowski

Status:	Accepted	**Start date:**	11/30/2015
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:	Infrastructure	**Estimated time:**	0.00 hour
Target version:	Capella	**Spent time:**	0.00 hour
Scope:	Internal	**SSI Package:**	
Experiment:	-		

Description

I would like to see the tuple product support that Jesse demonstrated over the summer available in an art release. The initial work is available in artextensions on branch feature/pymod, in directory artextensions/PyMod. The interface is defined here. One of the primary goals for this product is to support seamless integration with art modules written in Python (there will be other related issues for this).
This general Ntuple product has can he thought of as an "struct of arrays" products, with efficient storage and access by column, and support for row access (rows returned as std::tuple objects).
Permitting the general access in Python requires use of Event view support and an "ntuple" base class.

A few notes of things to keep in mind here:

1. ROOT 5.x could not support such a structured product directly.
2. C++11 variadic templates can support such a structure. This has been demonstrated. It is unclear whether the user experience will be good enough i.e. the instantiated class might be do horrible to deal with.
3. the demo system uses simple code generation to explicitly write the n-tuple product using a simple struct-like definition (description of one row).
4. ROOT 6 claims to be able to support such an C++ template structure, but specialized support for some some of the C++ containers that might be used will likely not be available, meaning that the guts of C++11 std classes might be visible in ROOT class definitions.

Here are the issues:

1. code generation is used in the artextensions area. A product called "cog" is used (a python code generation tool. The procedure will need to be looked over and reviewed.
2. view support requires a container to be available in the event. This n-tuple product is a first-class product, not living in a container. The current implementation requires an artificial container with one element to permit the view support to work. Views support for first-class products is needed for this to work cleanly.

Assuming a row that looks like this:

```
MyThing
 int x
 double y
```

Here is the approximate interface that is generated:

```
struct MyThing : public artext::NTupleInterface {
 std::vector<int> x;
 std::vector<double y;

 std::tuple<int,double> getRow(unsigned int index) const;
 void addRow(std::tuple<int,double> row);

 ... all the view support stuff
 private:
 ... all the stuff that artext::NTupleInterface requires to be implemented for generic access
};
```

Related issues:

Related to art - Feature #11036: multidimensional array support in NTuple pro...	**Accepted**	**11/30/2015**

History

#1 - 11/30/2015 04:52 PM - Jim Kowalkowski

- Description updated

#2 - 11/30/2015 05:19 PM - Jim Kowalkowski

- Precedes Feature #11033: art module in Python added

#3 - 11/30/2015 09:07 PM - Jim Kowalkowski

- Related to Feature #11035: art producer module in python added

#4 - 11/30/2015 09:07 PM - Jim Kowalkowski

- Related to deleted (Feature #11035: art producer module in python)

#5 - 11/30/2015 09:12 PM - Jim Kowalkowski

- Related to Feature #11036: multidimensional array support in NTuple product added

#6 - 12/07/2015 11:43 AM - Kyle Knoepfel

- Status changed from New to Accepted

#7 - 12/07/2015 11:43 AM - Kyle Knoepfel

- Target version set to 2.01.00

#8 - 05/06/2016 02:03 PM - Kyle Knoepfel

- Target version changed from 2.01.00 to Vega

#9 - 04/23/2018 10:51 AM - Kyle Knoepfel

- Target version changed from Vega to Capella