Ticket #4399 (closed Patches: fixed)

Make the free function 'size' support the uBLAS traits system and better work with expression types

Opened 2 years ago
Last modified 23 months ago

Reported by:	Marco Guazzone <marco.guazzone@…>	Owned by:	david.bellot
Milestone:	~~Boost 1.45.0~~	Component:	uBLAS
Version:	Boost Development Trunk	Severity:	Problem
Keywords:		Cc:	

Description

Actually the implementation of the **size** free function has two weak points:

1. It does not make use of the new uBLAS traits system introduced by Gunter.
   E.g.: *typename ExprT::size_type* should become *typename matrix_traits<ExprT>::size_type*
2. It does not fully support expression types. The current support is error prone and not user-friendly.
   E.g.: if *ExprT* is an expression type and *e* is a variable of type *ExprT*, then one has to call *size(e())* instead of simply *size(e)*.

I suggest to change the **size** operation in order to fix the above issues.

Attachments

- size-allow_expr_and_break_back_comp.patch (32.1 KB) - added by *Marco Guazzone <marco.guazzone@…>* 2 years ago.
  Adds the use of uBLAS type traits, simplifies the interaction with expression types, replaces the 'size<tag>' function with the 'size_by_tag<tag>' function.
- size-allow_expr.patch (18.7 KB) - added by *Marco Guazzone <marco.guazzone@…>* 2 years ago.
  Patch for size.hpp which is back compatible.

Change History

Changed 2 years ago by Marco Guazzone <marco.guazzone@…>

- **attachment** *size-allow_expr_and_break_back_comp.patch* added

  Adds the use of uBLAS type traits, simplifies the interaction with expression types, replaces the 'size<tag>' function with the 'size_by_tag<tag>' function.

Changed 2 years ago by Marco Guazzone <marco.guazzone@…> comment:1

I've tried to change the 'size' operation without affecting its syntax but I've failed.

More specifically, the problem is given by one of the polymorphic version of *size*:

```
template <typename TagT, typename ExprT>
typename ExprT::size_type size(ExprT const& e);
```

which I've initially transformed into

```
template <typename TagT, typename MatrixExprT>
typename matrix_traits<MatrixExprT>::size_type size(matrix_expression<
```

See the thread http://lists.boost.org/MailArchives/ublas/2010/06/4362.php for more details.

So, my solution is to **break back compatibility** by removing that version of **size** and introducing a new free function (e.g., **size_by_tag**) with the same semantic.

I attach a patch for the **size** operation and the related test suite.

Changed 2 years ago by Marco Guazzone <marco.guazzone@...>

- **attachment** *size-allow_expr.patch* added

Patch for size.hpp which is back compatible.

Changed 2 years ago by Marco Guazzone <marco.guazzone@...> comment:2

At the end I was able to create a patch for size.hpp which does not break back compatibility. So ignore the initially submitted patch (*size-allow_expr_and_break_back_comp.patch*).

I admit the real credits go to Daniel & Stefan, two guys that suggested to me the way to go. For more info see: http://groups.google.com/group/comp.lang.c++.moderated/browse_thread/thread/bd5080b28865f826

Summary of changes:

- Use of ublas type-traits system (e.g. typename matrix_traits<M>::size_type).
- Explicit use of matrix_/vector_expression in function arguments (e.g. size(matrix_expression<M> const& m).
- Call to size<1>(v), with v a vector expression, is legal and returns the length of the vector-
  Do you like it?

Changed 23 months ago by anonymous comment:3

patch applied and commited in trunk

Changed 23 months ago by david.bellot comment:4

- **Owner** changed from *guwi17* to *david.bellot*
- **Version** changed from *Boost 1.44.0* to *Boost Development Trunk*
- **Milestone** changed from *Boost 1.44.0* to *Boost-1.45.0*

Changed 23 months ago by david.bellot comment:5

- **Status** changed from *new* to *closed*
- **Resolution** set to *fixed*