P1709 / graph-v2 Identifiers

## Identifier definition

An abstract and opaque value that is used by functions to do their work.

Separate vertex and edge identifiers replace the use of references. For vertices, the identifier also removes the requirement for a vertex id.

### Benefits

* Simplifies the interface
  + Overloaded GCI functions can be consolidated into a single function.
  + Collapse “basic” and non-basic concepts into a single definition.
  + Collapse “basic” and non-basic views into a single view.
* Improves the possibility of having a single algorithm that can work on random-access and bidirectional vertex containers.

### Key Interface Changes

* GCI
  + Vertex functions
    - find\_vertex(g,uid) 🡪 ~~vertex\_iterator\_t<G>~~ vertex\_identifier\_t<G>
    - vertices(g) 🡪 range of identifiers? (specialized range)
    - vertex\_exists(g,uid) (new function)
    - Vertex id and/or reference parameters will be consolidated to a vertex identifier.
    - Vertex id and vertex identifier are distinct concepts (a vertex identifier could be a vertex id)
  + Edge functions
    - edges(g,ui) 🡪 range of edge identifiers? (specialized range)
    - find\_outgoing\_edge(g,vertex\_identifier,vertex\_identifier) -> edge\_identifier
    - find\_outgoing\_edge(g,vertex\_id,vertex\_id) -> edge\_identifier
    - edge\_exsits(g,uvid) (new function)
    - contains\_edge(g,vertex\_identifier,vertex\_identifier) -> bool
    - contains\_edge(g,vertex\_id,vertex\_id) -> bool
    - Vertex id and vertex reference parameters will be replaced by a vertex identifier.
    - Edge reference parameters will be replaced by edge identifier.
* Views
  + Vertex id and vertex reference parameters will be consolidated to a single vertex identifier.
  + Edge references will be replaced by edge identifiers.
  + (Consider addition of additional member in descriptors for events)
* Algorithms
  + Vertex ids will remain
  + Range of vertex ids will remain.
  + Additional overload for range of vertex identifiers?

### Implementation

#### Vertex identifier

Obvious implementations: integral index, iterator

Requirements

* find\_vertex(g,vertex id) 🡪 vertex\_identifier\_t<G>
* vertex\_id(g,vid) 🡪 vertex\_id\_t<G>

#### Edge identifier

Obvious implementations: integral index (adjacency matrix), iterator

### Terminology?

Vertex id vs. vertex identifier

Descriptor vs identifier