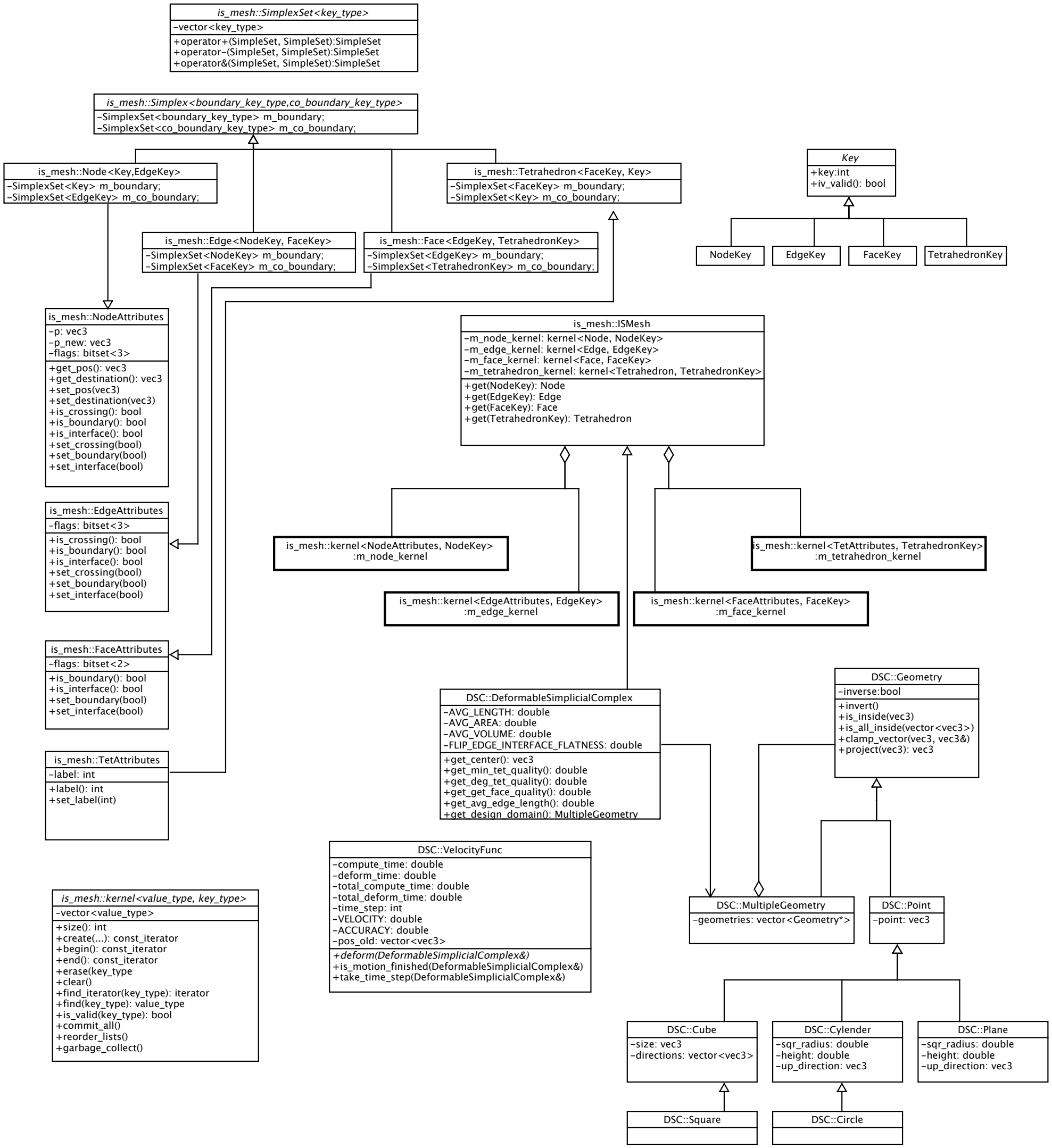
is_mesh::SimplexSet<key_type>
-vector<key_type>
+operator+(SimpleSet, SimpleSet):SimpleSet
+operator-(SimpleSet, SimpleSet):SimpleSet
+operator&(SimpleSet, SimpleSet):SimpleSet
is_mesh::Simplex<boundary_key_type,co_boundary_key_type>
-SimplexSet<boundary_key_type> m_boundary;
-SimplexSet<co_boundary_key_type> m_co_boundary;
is_mesh::Node<Key,EdgeKey>
-SimplexSet<Key> m_boundary;
-SimplexSet<EdgeKey> m_co_boundary;
is_mesh::Edge<NodeKey, FaceKey>
-SimplexSet<NodeKey> m_boundary;
-SimplexSet<FaceKey> m_co_boundary;
is_mesh::Face<EdgeKey, TetrahedronKey>
-SimplexSet<EdgeKey> m_boundary;
-SimplexSet<TetrahedronKey> m_co_boundary;
is_mesh::Tetrahedron<FaceKey, Key>
-SimplexSet<FaceKey> m_boundary;
-SimplexSet<Key> m_co_boundary;
Key
+key:int
+iv_valid(): bool
NodeKey
EdgeKey
FaceKey
TetrahedronKey
is_mesh::NodeAttributes
-p: vec3
-p_new: vec3
-flags: bitset<3>
+get_pos(): vec3
+get_destination(): vec3
+set_pos(vec3)
+set_destination(vec3)
+is_crossing(): bool
+is_boundary(): bool
+is_interface(): bool
+set_crossing(bool)
+set_boundary(bool)
+set_interface(bool)
is_mesh::EdgeAttributes
-flags: bitset<3>
+is_crossing(): bool
+is_boundary(): bool
+is_interface(): bool
+set_crossing(bool)
+set_boundary(bool)
+set_interface(bool)
is_mesh::FaceAttributes
-flags: bitset<2>
+is_boundary(): bool
+is_interface(): bool
+set_boundary(bool)
+set_interface(bool)
is_mesh::TetAttributes
-label: int
+label(): int
+set_label(int)
is_mesh::ISMesh
-m_node_kernel: kernel<Node, NodeKey>
-m_edge_kernel: kernel<Edge, EdgeKey>
-m_face_kernel: kernel<Face, FaceKey>
-m_tetrahedron_kernel: kernel<Tetrahedron, TetrahedronKey>
+get(NodeKey): Node
+get(EdgeKey): Edge
+get(FaceKey): Face
+get(TetrahedronKey): Tetrahedron
is_mesh::kernel<NodeAttributes, NodeKey>
:m_node_kernel
is_mesh::kernel<EdgeAttributes, EdgeKey>
:m_edge_kernel
is_mesh::kernel<FaceAttributes, FaceKey>
:m_face_kernel
is_mesh::kernel<TetAttributes, TetrahedronKey>
:m_tetrahedron_kernel
is_mesh::kernel<value_type, key_type>
-vector<value_type>
+size(): int
+create(...): const_iterator
+begin(): const_iterator
+end(): const_iterator
+erase(key_type
+clear()
+find_iterator(key_type): iterator
+find(key_type): value_type
+is_valid(key_type): bool
+commit_all()
+reorder_lists()
+garbage_collect()
DSC::DeformableSimplicialComplex
-AVG_LENGTH: double
-AVG_AREA: double
-AVG_VOLUME: double
-FLIP_EDGE_INTERFACE_FLATNESS: double
+get_center(): vec3
+get_min_tet_quality(): double
+get_deg_tet_quality(): double
+get_get_face_quality(): double
+get_avg_edge_length(): double
+get_design_domain(): MultipleGeometry
DSC::VelocityFunc
-compute_time: double
-deform_time: double
-total_compute_time: double
-total_deform_time: double
-time_step: int
-VELOCITY: double
-ACCURACY: double
-pos_old: vector<vec3>
+deform(DeformableSimplicialComplex&)
+is_motion_finished(DeformableSimplicialComplex&)
+take_time_step(DeformableSimplicialComplex&)
DSC::Geometry
-inverse:bool
+invert()
+is_inside(vec3)
+is_all_inside(vector<vec3>)
+clamp_vector(vec3, vec3&)
+project(vec3): vec3
DSC::MultipleGeometry
-geometries: vector<Geometry*>
DSC::Point
-point: vec3
DSC::Cube
-size: vec3
-directions: vector<vec3>
DSC::Cylender
-sqr_radius: double
-height: double
-up_direction: vec3
DSC::Plane
-sqr_radius: double
-height: double
-up_direction: vec3
DSC::Square
DSC::Circle