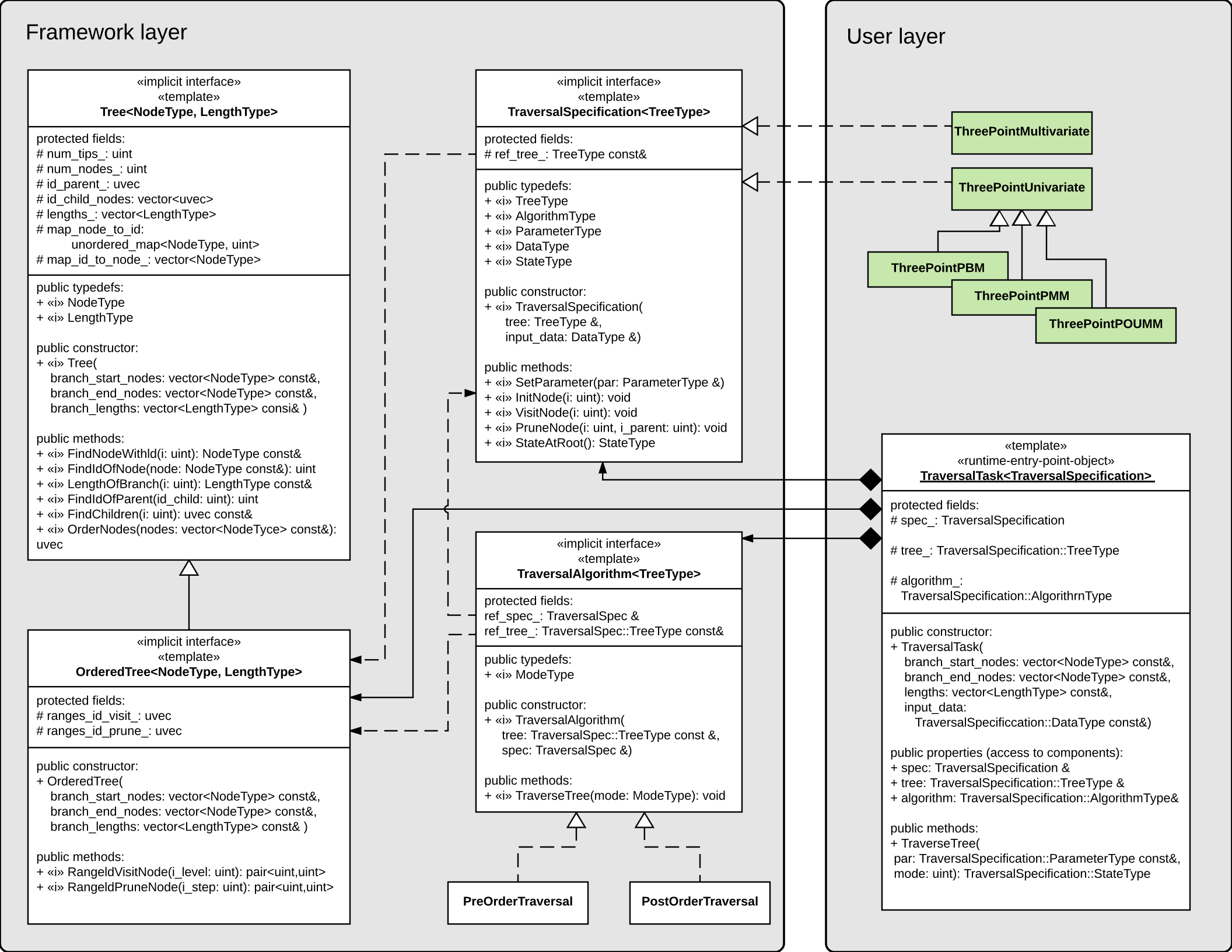
Framework layer
User layer
«implicit interface»
«template»
Tree<NodeType, LengthType>
protected fields:
num_tips_: uint
num_nodes_: uint
id_parent_: uvec
id_child_nodes: vector<uvec>
lengths_: vector<LengthType>
map_node_to_id:
unordered_map<NodeType, uint>
map_id_to_node_: vector<NodeType>
public typedefs:
+ «i» NodeType
+ «i» LengthType
public constructor:
+ «i» Tree(
branch_start_nodes: vector<NodeType> const&,
branch_end_nodes: vector<NodeType> const&,
branch_lengths: vector<LengthType> consi&)
public methods:
+ «i» FindNodeWithId(i: uint): NodeType const&
+ «i» FindIdOfNode(node: NodeType const&): uint
+ «i» LengthOfBranch(i: uint): LengthType const&
+ «i» FindIdOfParent(id_child: uint): uint
+ «i» FindChildren(i: uint): uvec const&
+ «i» OrderNodes(nodes: vector<NodeTyce> const&): uvec
«implicit interface»
«template»
TraversalSpecification<TreeType>
protected fields:
ref_tree_: TreeType const&
public typedefs:
+ «i» TreeType
+ «i» AlgorithmType
+ «i» ParameterType
+ «i» DataType
+ «i» StateType
public constructor:
+ «i» TraversalSpecification(
tree: TreeType &,
input_data: DataType &)
public methods:
+ «i» SetParameter(par: ParameterType &)
+ «i» InitNode(i: uint): void
+ «i» VisitNode(i: uint): void
+ «i» PruneNode(i: uint, i_parent: uint): void
+ «i» StateAtRoot(): StateType
ThreePointMultivariate
ThreePointUnivariate
ThreePointPBM
ThreePointPMM
ThreePointPOUMM
«template»
«runtime-entry-point-object»
TraversalTask<TraversalSpecification>
protected fields:
spec_: TraversalSpecification
tree_: TraversalSpecification::TreeType
algorithm_:
TraversalSpecification::AlgorithrnType
public constructor:
+ TraversalTask(
branch_start_nodes: vector<NodeType> const&,
branch_end_nodes: vector<NodeType> const&,
lengths: vector<LengthType> const&,
input_data:
TraversalSpecificcation::DataType const&)
public properties (access to components):
+ spec: TraversalSpecification &
+ tree: TraversalSpecification::TreeType &
+ algorithm: TraversalSpecification::AlgorithmType&
public methods:
+ TraverseTree(
par: TraversalSpecification::ParameterType const&,
mode: uint): TraversalSpecification::StateType
«implicit interface»
«template»
TraversalAlgorithm<TreeType>
protected fields:
ref_spec_: TraversalSpec &
ref_tree_: TraversalSpec::TreeType const&
public typedefs:
+ «i» ModeType
public constructor:
+ «i» TraversalAlgorithm(
tree: TraversalSpec::TreeType const &,
spec: TraversalSpec &)
public methods:
+ «i» TraverseTree(mode: ModeType): void
«implicit interface»
«template»
OrderedTree<NodeType, LengthType>
protected fields:
ranges_id_visit_: uvec
ranges_id_prune_: uvec
public constructor:
+ OrderedTree(
branch_start_nodes: vector<NodeType> const&,
branch_end_nodes: vector<NodeType> const&,
branch_lengths: vector<LengthType> const&)
public methods:
+ «i» RangeIdVisitNode(i_level: uint): pair<uint,uint>
+ «i» RangeIdPruneNode(i_step: uint): pair<uint,uint>
PreOrderTraversal
PostOrderTraversal