Template Class FlowGraph

Table of contents

- Defined in File flow_graph.hpp

Inheritance Relationships

Base Type

- public holoscan::Graph< OperatorNodeType, OperatorEdgeDataElementType >
(Template Class Graph)

Class Documentation

template<typename NodeT = OperatorNodeType, typename EdgeDataElementT = OperatorEdgeDataElementType>
class FlowGraph : public holoscan::Graph<OperatorNodeType, OperatorEdgeDataElementType>

Public Types

using NodeType = NodeT

using NodePredicate = std::function<bool(const NodeType&)>

using EdgeDataElementType = EdgeDataElementT

using EdgeDataType = std::shared_ptr<EdgeDataElementType>

Public Functions

~FlowGraph() override = default

virtual void add_node(const NodeType &node) override

Add the node to the graph.

Parameters

node – The node to add.

void add_flow(const NodeType &node_u, const NodeType &node_v, const EdgeDataType &port_map) override

virtual std::optional<EdgeDataType> get_port_map(const NodeType &node_u, const NodeType &node_v) override

Get a mapping from the source node's port name to the destination node's port name(s).

Parameters

- **node_u** – A source node.
- **node_v** – A destination node.

Returns

A map from the source node's port name to the destination node's port name(s).

virtual bool is_root(const NodeType &node) override

Check if the node is a root node.

Parameters

node – A node in the graph.

Returns

true if the node is a root node.

inline virtual bool is_user_defined_root(const NodeType &node)

Check if the node is a user-defined root node. A user-defined root is the first node that is added to the graph.

Parameters

node – A node in the graph.

Returns

true if the node is a user-defined root node.

virtual bool is_leaf(const NodeType &node) override

Check if the node is a leaf node.

Parameters

node – A node in the graph.

Returns

true if the node is a leaf node.

virtual std::vector<NodeType> has_cycle() override

Returns a vector of root nodes of the cycles if the graph has cycle(s). Otherwise, an empty vector is returned.

Returns

Returns a vector of root nodes of cycles.

virtual std::vector<NodeType> get_root_nodes() override

Get all root nodes.

Returns

A vector of root nodes.

virtual std::vector<NodeType> get_nodes() override

Get all nodes.

The nodes are returned in the order they were added to the graph.

Returns

A vector of all nodes.

virtual std::vector<NodeType> get_next_nodes(const NodeType &node) override

Get the next nodes of the given node.

Parameters

node – A node in the graph.

Returns

A vector of next nodes.

virtual std::vector<NodeType> get_previous_nodes(const NodeType &node) override

Get the previous nodes of the given node.

Parameters

op – A node in the graph.

Returns

A vector of next nodes.

NodeType find_node(const NodePredicate &pred) override

virtual NodeType find_node(const NodeType &node) override

Find a node in the graph that is equal to the given node.

Parameters

node – The node to find.

Returns

The node in the graph if found, otherwise nullptr.

virtual NodeType find_node(std::string name) override

Find a node in the graph whose name is equal to the given name.

Parameters

name – The name to find.

Returns

The node in the graph if found, otherwise nullptr.