

WriterReaderAutomata

The WriterReaderAutomata class has four states:

READ_SIZE: In this state, the automata is waiting to read the size of the next message from the channel.

READ_CONTENT: In this state, the automata is waiting to read the content of the next message from the channel.

WRITE_SIZE: In this state, the automata is waiting to write the size of the next message to the channel.

WRITE_CONTENT: In this state, the automata is waiting to write the content of the next message to the channel.

The automata uses a WriterReaderListener object to receive notifications from the channel. When the channel is ready to read or write, the WriterReaderListener will call the appropriate method on the automata.

read: This method is called when the channel is ready to read data. It reads the next byte from the channel and determines what to do next based on the current state of the automata.

write: This method is called when the channel is ready to write data. It writes the next byte to the channel and determines what to do next based on the current state of the automata.

start: This method starts the automata. It calls the read method to start reading data from the channel.

CMessageQueue

The CMessageQueue class has four methods:

send: This method sends a message to the queue. It creates a new WriterReaderAutomata object and calls its write method to write the message to the channel.

setListener: This method sets a listener to receive notifications when messages are received and called the read method on the WriterReaderAutomata object.

close: This method closes the queue.

The CMessageQueue class uses the WriterReaderAutomata class to send and receive messages. When the send method is called, the CMessageQueue class will call the write method on the WriterReaderAutomata class. When the receive method is called, the CMessageQueue class will call the read method on the WriterReaderAutomata class.

CQueueBroker

The CQueueBroker class has three methods:

bind: This method binds to a port and accepts new connections. It calls the accept method on the broker object to start accepting new connections.

unbind: This method unbinds from a port. It calls the disconnect method on the broker object to stop accepting new connections.

connect: This method connects to a remote broker. It calls the connect method on the broker object to connect to the remote broker.

The CQueueBroker class uses the broker object to accept new connections and to connect to remote brokers. When a new connection is accepted, the CQueueBroker class will create a new CMessageQueue object and pass it to the AcceptListener object. When a connection is made to a remote broker, the CQueueBroker class will create a new CMessageQueue object and pass it to the ConnectListener object.

Event Handling

The code uses an event-driven architecture to handle communication between the components. When the channel is ready to read or write, the WriterReaderListener object will call the appropriate method on the WriterReaderAutomata class. When the WriterReaderAutomata class has finished reading or writing a message, it will call the received method on the listener object.