

State Machine

Document #: P2284R0
Date: 2021-10-29
Project: Programming Language C++
Audience: LEWGI
Reply-to: Ran Regev
<regev.ran@gmail.com>

Contents

1	Motivation and Scope	1
2	Terminology	1
2.1	FSM	1
2.2	State	1
2.3	Transition	2
2.4	Context	2
3	Open Issues	2
4	Sources	2
5	Acknowledgements	2

1 Motivation and Scope

State Machines are fundamental aspect of computer science and are widely used in the industry. There are many aspects to consider when implementing a state machine framework, and a good standard library can ease the burden for developers.

2 Terminology

State Machine means different things to different people. This section sets the terminology for the rest of the paper.

2.1 FSM

FSM is a Finite State Machine. It encapsulates everything this proposal suggests. The facility this paper proposed is called fsm.

```
std::fsm my_fsm;
```

A FSM has the following responsibilities: * Holds its states * Holds a common context for all states * Transit between states.

2.2 State

A State is a position in the state machine. The state machine itself can rest, at a given time, only in a single state.

2.3 Transition

A Transition is the move of the state machine from one state to another.

2.4 Context

The context of a state machine is the common data or functionality that is accesible to all states.

```
std::fsm::state wait_for_input("wait-for-input");
std::fsm::state open("open");
std::fsm::state locked("locked");

std::fsm::transition(wait_for_input, open, [](const std::string password) -> std::fsm::state& {
    if (fsm::context::retries > 3) {return locked;}
    if (password_ok(password)) {return open;}
    ++fsm::context::retries;
    return wait_for_input
}
int retries = 0;
std::fsm<int> myfsm(retries);
```

3 Open Issues

- Completeness: should all states handle all inputs? ** Yes: it is safer, it covers all possible situations. **
No: it is just too much effort from the developer point of view to force declarig handling of impossible combinations of state/input.
- Explicit and enforced states transitions vs. implicit and non enforced state transition ADD EXAMPLE

4 Sources

The code for the diagrams in this paper are witten in PlantUML and can be used to regenrate the drawing with uml-generator like <https://www.planttext.com>

5 Acknowledgements

Michael Park mcpark@gmail.com (for github.com/mpark/wg21)