State Machine

Document #:	PXXXXR0
Date:	2021-02-16
Project:	Programming Language C++
Audience:	LEWGI

Reply-to:	Ran Regev <regev.ran@gmail.com>

Contents

1 Motivation and Scope

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

2 Terminology

State Machine means different things to different people. This sectoin 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;
```

2.2 State

A State is ...

2.3 Transition

A Transition is ...

3 Types Of State Machines

There are two main types of state machines in use * “Academic” * Industrial

3.1 “Academic”

The academic state machine works on its input and stops on a state. The main interest in this type of fsm is its *final state*. By investigating the fsm’s final state the user knows the answer to the question being asked. For example, for words built from the {a,b} alphabiet deciding if a string has odd number of ’a’s one can build a fsm that its final state answer this question.

```
bool odd_a(const std::string word) {
    std::fsm my_fsm = /*...*/ // construct the fsm
    for ( char c : word ) {
        my_fsm.fire(c);
    }
    return my_fsm.state() == odd;
}
```

3.2 Industrial

The industrial state machine normally works forever and the main interest is in the behavior in each state regarding the input. For example, a three tries password access can be moduled with fsm like:

```
    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) -> bool {return password_ok
    std::fsm::transition(w
```

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)