

C++ NOW 2017

**[BOOST].SML**

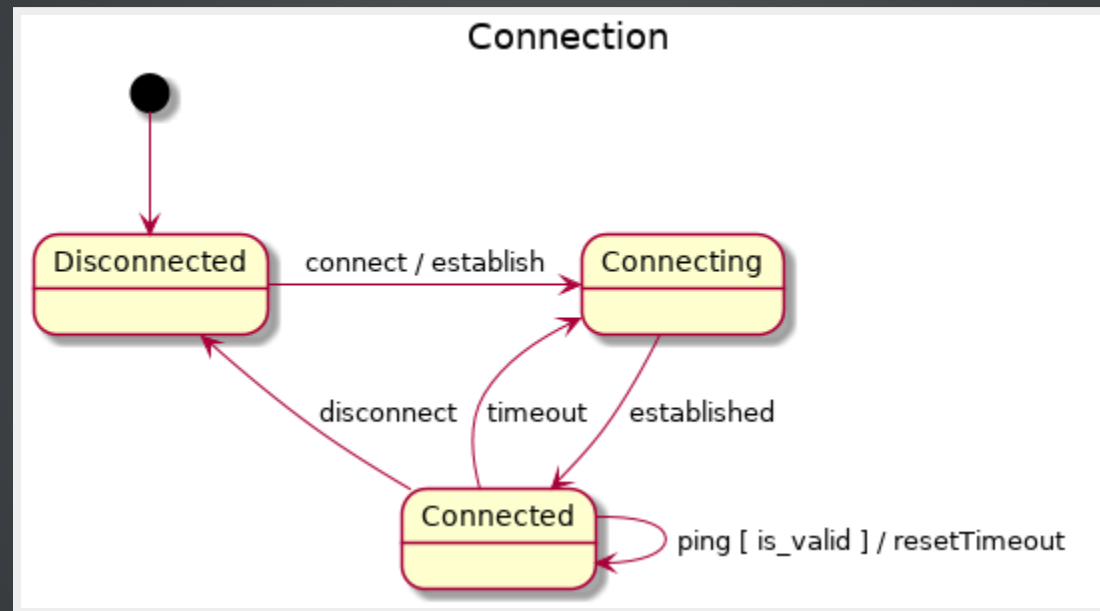
# STATE MACHINE LANGUAGE

Kris Jusiak, Quantlab Financial

---

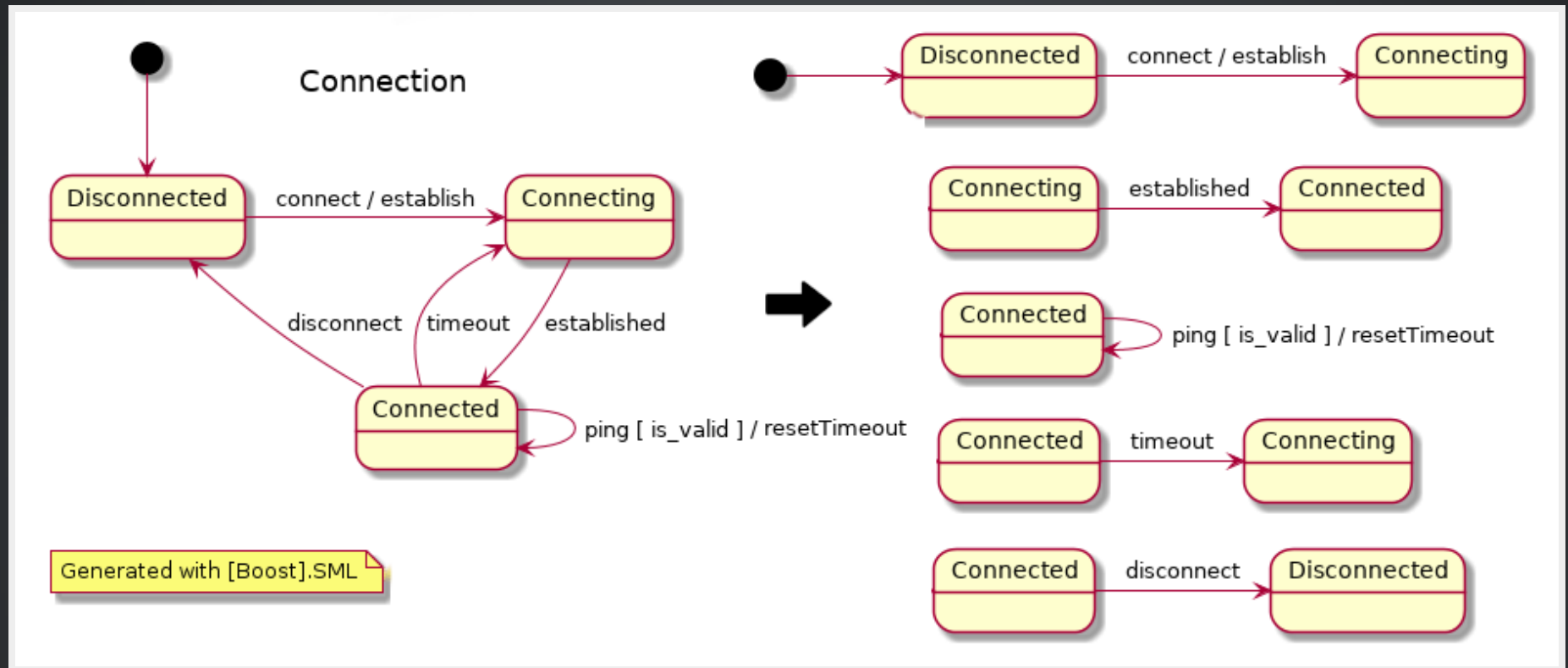
[kris@jusiak.net](mailto:kris@jusiak.net) | [@krisjusiak](https://twitter.com/krisjusiak) | [linkedin.com/in/kris-jusiak](https://www.linkedin.com/in/kris-jusiak)

# CONNECTION - STATE DIAGRAM

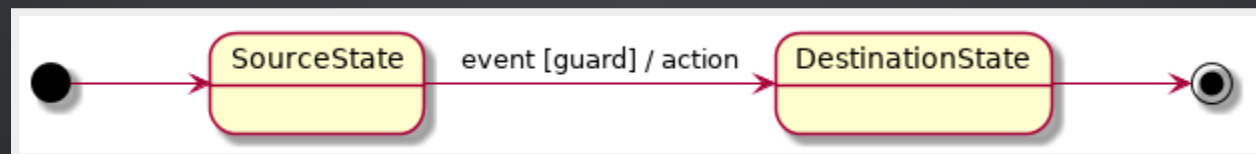


UNIFIED MODELING LANGUAGE (UML 2.5)

# CONNECTION - STATE DIAGRAM - TRANSITION



## TRANSITION



# [BOOST].SML - CONNECTION

```
// guards
const auto is_valid = [] (auto event) { return event.valid; };

// actions
const auto establish = [] { std::puts("establish"); };
const auto close = [] { std::puts("close"); };
const auto resetTimeout = [] { std::puts("resetTimeout"); };

// events (types)
struct connect {}; // ...
struct ping { bool valid{}; };
```

```
int main() {
    sm::sm connection = [] {
        return transition_table{
            * "Disconnected"_s + connect / establish           = "Connecting"_s,
            "Connecting"_s    + established                     = "Connected"_s,
            "Connected"_s     + ping [ is_valid ] / resetTimeout,
            "Connected"_s     + timeout / establish             = "Connecting"_s,
            "Connected"_s     + disconnect / close              = "Disconnected"_s
        };
    };

    sm.process_event(connect{});
}
```

# [BOOST].SML - PERFORMANCE

Compiler Explorer

C++ Editor Diff View More

Share Help

C++ source #1

```
12 const auto close = []{ std::puts("close"); };
13 const auto is_valid = [](const auto& event) { return event.valid; };
14 const auto resetTimeout = [] { std::puts("resetTimeout"); };
15
16 struct Connection {
17     auto operator()() const {
18         using namespace sml;
19         return make_transition_table(
20             * "Disconnected"_s + event<connect> / establish = "Connecting"_s,
21             "Connecting"_s + event<established> = "Connected"_s,
22             "Connected"_s + event<ping> [ is_valid ] / resetTimeout,
23             "Connected"_s + event<timeout> / establish = "Connecting"_s,
24             "Connected"_s + event<disconnect> / close = "Disconnected"_s
25         );
26     }
27 };
28
29 int main() {
30     sml::sm<Connection> connection{};
31     connection.process_event(connect{});
32     connection.process_event(established{});
33     connection.process_event(ping{true});
34     connection.process_event(disconnect{});
35 }
36
37
38
39
40
```

x86-64 clang 4.0.0 (Editor #1, Compiler #1)

x86-64 clang 4.0.0 -O3 -DNDEBUG -std=c++1z

11010 .LX0: .text // Intel A

```
1 main: # @main
2     pushq    %rax
3     movl     $.L.str, %edi
4     callq    puts
5     movl     $.L.str.1, %edi
6     callq    puts
7     movl     $.L.str.2, %edi
8     callq    puts
9     xorl     %eax, %eax
10    popq     %rcx
11    retq
12
13 .L.str:
14     .asciz   "establish"
15
16 .L.str.1:
17     .asciz   "resetTimeout"
18
19 .L.str.2:
20     .asciz   "close"
```

clang version 4.0.0 (tags/RELEASE\_400/final 299826)-cached

# BENCHMARK - RESULTS

	Enum/Switch	Variant	[Boost].SML	Boost.MSM-eUML	Boost.Statechart
Compilation time	0.132s	15.321s	0.582s	1m15.935s	5.671s
Execution time	679ms	827ms	622ms	664ms	2282ms
Memory usage	1b	2b/8b	1b	120b	224b
Executable size	15K	187K	34K	611K	211K
Line of Code (LOC)	~300 (no macros)	~300	~50	~100	~200

# [BOOST].SML / QUESTIONS?

- One header - 2k LOC - (boost/sml.hpp) / generated
- Neither Boost nor STL is required
- Quick compilation-times (-Wall -Wextra -Werror -pedantic -pedantic-errors)
- Blazing fast run-time (Generated at compile-time)
- No 'virtual's (-fno-rtti)
- Optional support for 'exception's (-fno-exceptions)
- Supported compilers (C++14)
  - Clang-3.4+, XCode-6.1+, GCC-5.2+, MSVC-2015+

---

<https://github.com/boost-experimental/sml>