ACCU 2017

Bristol, April 25th-29th

Trip Report Felix Petriconi

My Trip Report

- 3rd visit of the ACCU conference
- 2nd conference with a contribution by myself
- 1st conference as a conference committee member

Finding the Program

- ~120 sessions and tutorials were proposed
- 4 pre-conference tutorials and 60 sessions had to be chosen
- Meeting in London on December the 3rd

ACCU - Association of C and C++ Users

- ACCU organization completely based on volunteers help
- 21th ACCU conference
- The complete spillover goes to Code Club, an charity organization that offers *all* children the possibility to learn to code

My Pre-Conference Tutorial

The Art of Writing Reasonable Concurrent Code

- Recap why writing concurrent programs is necessary today
- Introduction into high level concurrency abstractions
- Introduction into C++11 standard futures

My Pre-Conference Tutorial Cont.

- Introduction into C++17 (TS) / boost future extensions
- Showing the deficiencies of the current C++ standard's design
- In depth tutorial with exercises into an alternative future implementation by Sean Parent from Adobe and myself: http://www.stlab.cc
- In depth tutorial with exercises into CSP channels.

Conference Highlights 1st day

- Keynote by Russ Miles
  - About the „Highway to Hell“ of micro-service architecture
- Nico Josuttis: The nightmare of Move Semantics for Simple Classes
- Louis Dionne: Metaprogramming in C++14
  - Introduction into boost::hana, a combination of boost::mpl and boost::fusion

Marshall Clow

The Detection Idiom - a simpler way to SFINAE in C++17*

Pre-Detection Idiom Way

```
template <typename T>
  class has_PrintTo {
    typedef char   Yes;
    typedef struct No { char dummy[2]; };

    static std::ostream os;

    template<typename U>
    static auto test(U* p) ->
      decltype(PrintTo(*p, &os), Yes(0));

    template<typename>
    static No test(...);
  public:
    static const
      bool value =
        sizeof(test<T>(nullptr)) == sizeof(Yes);
  };
```

New Detection Idiom Way

```
template<typename T>
using print_to_t = decltype(
    PrintTo(std::declval<T&>(), std::declval<*std::ostream>()
);

template<typename T>
constexpr bool has_print_to_v = is_detected_v<print_to_t, T>;
```

* The code from proposal N4502 code can already be used with C++14, Visual Studio 2015

Conference Highlights 2nd day

- Keynote by Brad Chamberlain from Cray Inc.
    - Current state of the open source language Chapel for HPC
- John Lakos: Local (arena) Memory Allocators
    - Introduction into the new allocator concepts that come with C++17
    - Overview of performance gain in different contexts
- Sergey Ignatchenko: Deterministic Components for Interactive Distributed Systems: Benefits and Implementation
    - Definition of (non-) deterministic systems
    - Overview of how to separate in a system deterministic from non deterministic components
    - Demonstration of debugging techniques in a massive multi-player game
- Michel Grootjans: Why limiting your work in progress will speed up your project

Bloomberg Event

```
# a new thread
# it evals the code below...
# that thread doesn't interfere with other

# so that was lots of threads spawned...
# you're hearing them all at once.....

# i can scale samples up and down with midi pit
with_fx :reverb, room: 1 do
  with_fx :bitcrusher, sample_rate: 3000, bits:
    sample bass_atmos, 2, amp: 5, rpitch: 6
    sample bass_atmos, 2, amp: 5, rpitch: 6, rate
  end
end
```

Images by Austin Bingham, Vittorio Romeo and Paul Williams

Conference Highlights 3rd day

- Keynote by Fran Buontempo
  - AI: Actual Intelligence
- Sergei Sadovnikov: Automatic C++ source code generation with clang
  - Introduction into clang compiler front end
  - Demonstration of how simple it is to generate source code for enum to string conversion or data serialization and deserialization code
- Dominic Robinson: Coroutines and C++ DSLs for Human Scale Concurrency
  - Basic introduction into co-routines C++17(TS) / Visual Studio 2015
  - Introduction into his new frame to easily compose in a type safe manner state machines

Conference Dinner

Image by Neil Horlock

Image by Austin Bingham

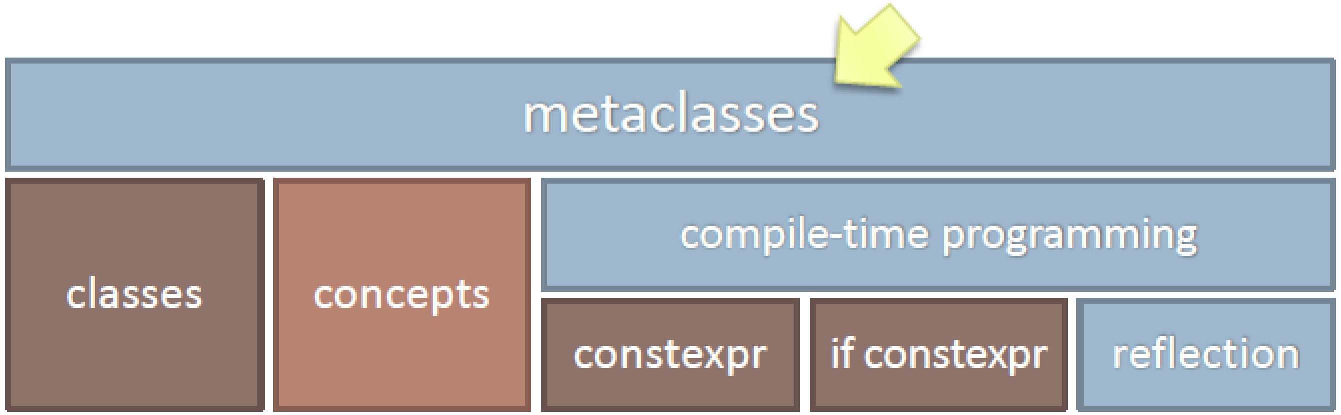
Conference Highlights 4th day

- Niall Douglas: Mongrel Monads, Dirty, Dirty, Dirty
  - Why one should favor expected<T,E> instead of error code and exceptions
- Vittorio Romeo: Implementing variant visitation using lambdas
- Odin Holmes: Modern C++ Design reloaded
  - Based on Alexandrescu‘s classic he revisited different techniques and showed how to achieve today more with simpler code

Image by Jakie Kay

Closing Keynote by Herb Sutter

- He spoke about a proposal to introduce Meta Classes into C++. Andrew Sutton and he are currently working on it
- First time mentioning in the public at ACCU. Therefore video publishing will be delayed until next standard meeting
- Based on the constexpr, if constexp, reflection



metaclasses

classes

concepts

compile-time programming

constexpr

if constexpr

reflection

Closing Keynote by Herb Sutter

```
class Shape {
public:
  virtual int area() const = 0;
  virtual void scale_by( double factor) = 0;
  // …
  ~virtual Shape() noexcept {}
};
```

```
interface Shape {
  int area() const;
  void scale_by(double factor)
  // …
};
```

Meta Classes for C++

```
$class interface{
  ~interface() noexcept{ }
  constexpr{
    compiler.require($interface.variables().empty(),
      "interfaces may not contain data members");
    for (auto f : $interface.functions()) {
      compiler.require(!f.is_copy() && !f.is_move(),
        "interfaces may not copy or move; consider a virtual clone()");
    if (!f.has_access())
      f.make_public();
    compiler.require(f.is_public(), "interface functions must be public");
    f.make_pure_virtual();
  }
}
```

Meta Classes: Qt Example

```
class MyClass: public QObject{
  Q_OBJECT
public:
  MyClass( QObject* parent = 0 );
  Q_PROPERTY(int value READ get_value
    WRITE set_value)
  int  get_value() const { return value; }
  void set_value(int v) { value = v; }
private:
  int value;
signals:
  void mySignal();
public slots:
  void mySlot();
};
```

```
QClass MyClass {
  property<int> value { };  // default
  signal mySignal();
  slot mySlot();
};
```

More on Meta

Fun with Reflection in C++ by Jakie Kay

(https://isocpp.org/blog/2017/05/fun-with-reflection-in-cpp-jakie-kay)

Online Compiler of Herb Sutter‘s and Andrew Sutton‘s work

(https://cppx.godbolt.org/)