

Systems Development with *AspeCt-oriented C (ACC)*

Presented by: *Michael Gong*

Joint work with Charles Zhang and Hans-Arno Jacobsen

University of Toronto



Crosscutting in C

```
void *NutHeapAlloc(size_t size) {
    #ifdef NUTDEBUG
        //code removed
    #endif
    NODE **fpp = 0;
    //code removed
    #if defined(__arm__) ||
        defined(__m68k__) ||
        defined(__H8300H__) || ...
        while ((size & 0x03) != 0)
            size++;
    #endif
    if (size >= available) {
        #ifdef NUTDEBUG
            //code removed
        #endif
        return 0;
    }
    //code removed
    ...(next column)
```

```
while (node) {
    //code removed
    if (fit) {
        //split the node if too big
        if (fit->hn_size > ...) {
            //code removed
            *fpp = node;
        } else
            *fpp = fit->hn_next;
    }
    //code removed
}
#ifdef NUTDEBUG
    //code removed
#endif
return fit; }
```

Nut/OS, heap.c

debug concern

system-specific concern

optimization concern

main functionality

Code Summary & Cause

- Certain concerns crosscut the core logic
 - a.k.a. *crosscutting concerns*
- Similar concern code **scatters** across the code
- Different pieces of concern code **tangled** with core
- Scattering, tangling, and crosscutting leads to
 - code -> *hard to read, understand and maintain*
 - design intent -> *not cleanly represented*
 - concerns -> *not well separated and modularized*

Aspect-Oriented Programming



■ AOP

- a programming paradigm
- modularization of crosscutting concerns in software
- **complementary** to existing paradigms: OO, ...

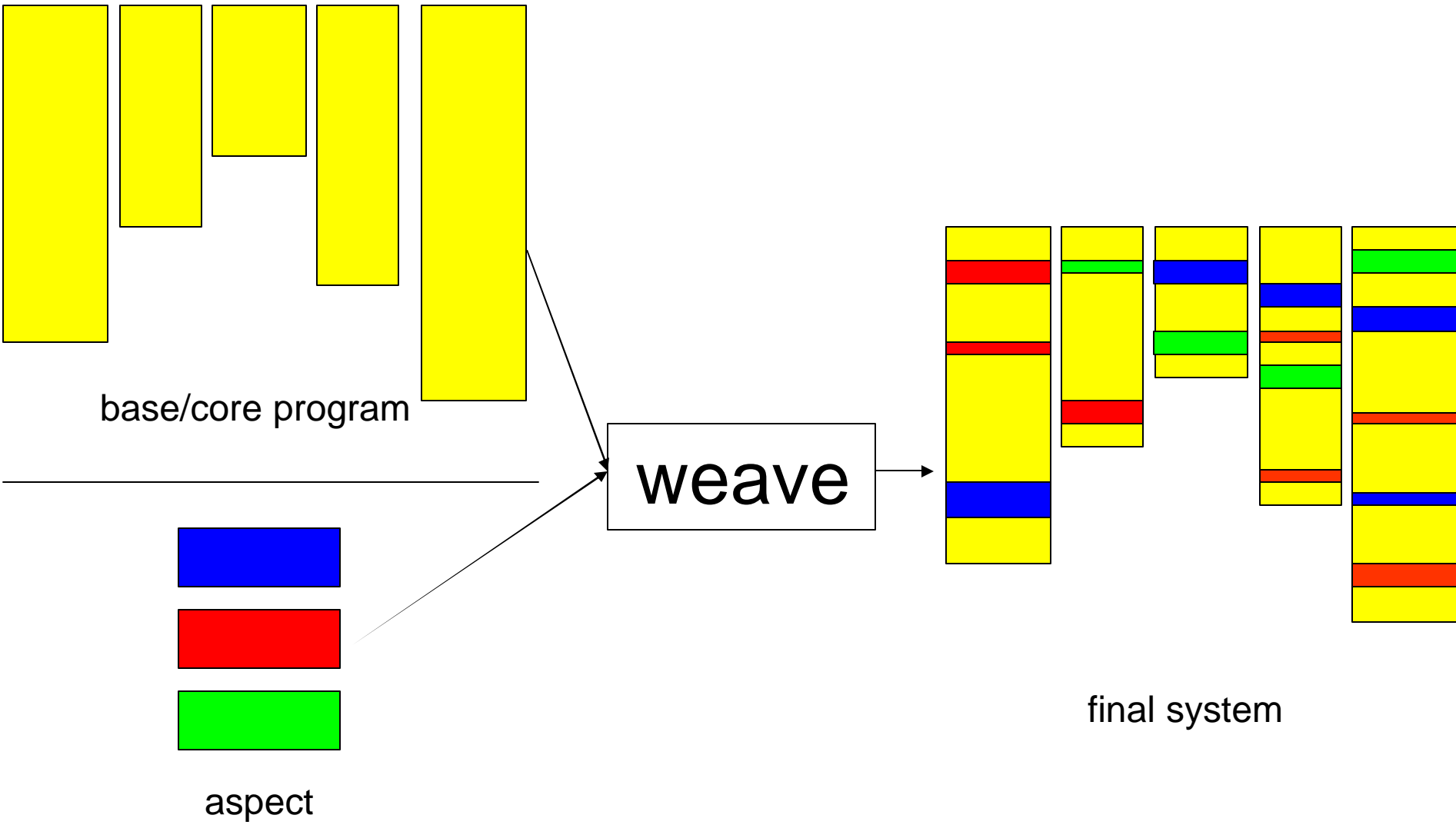
■ History

- about 10 years ago
- research efforts on separation of concerns in software

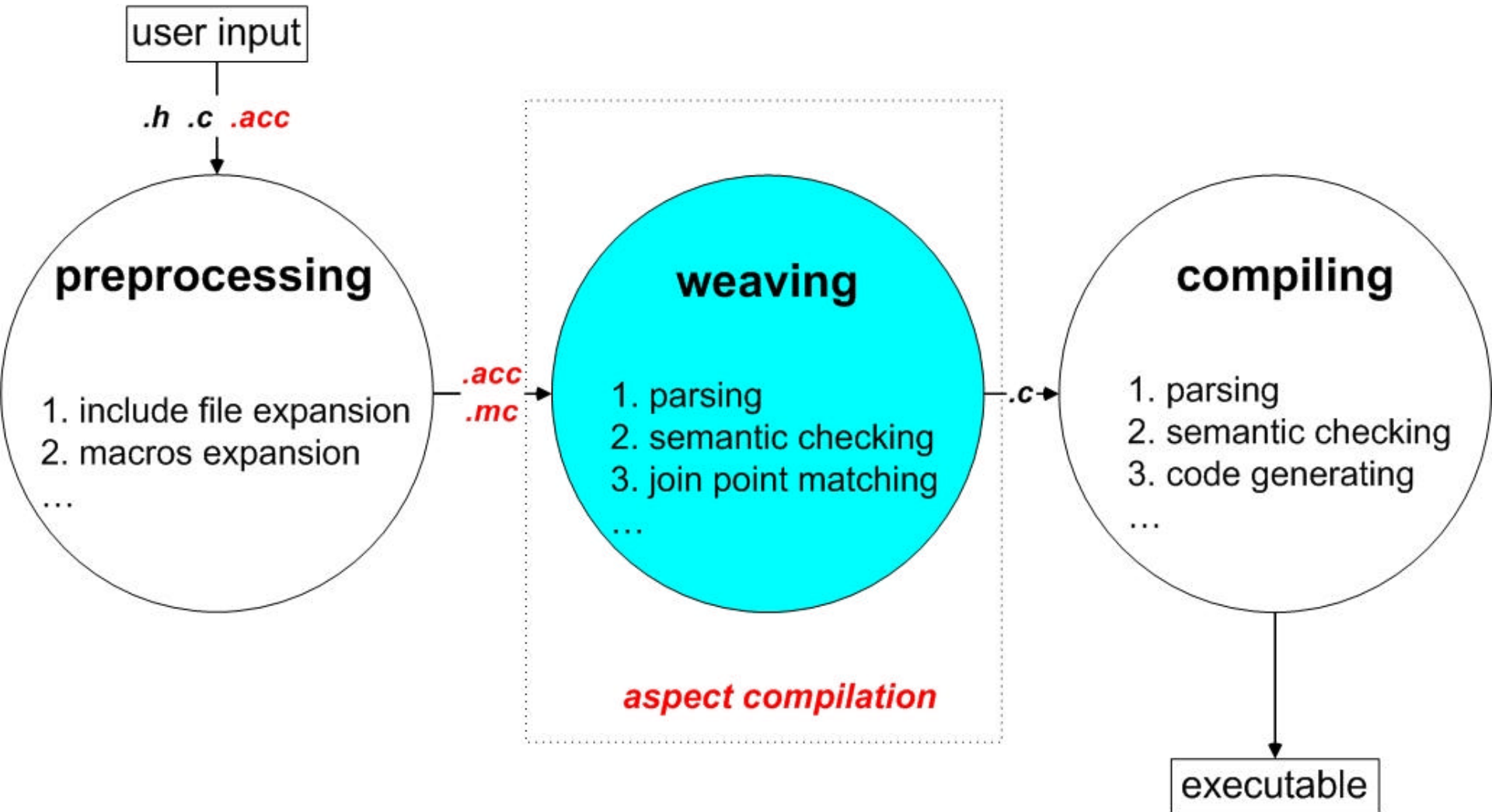
■ Now

- supported in industry today by IBM, BEA, ...
- AOP support is available for Java, C++, C, PHP, ...
 - AspectJ, AspectC++, AspectC, AOPHP, ...

AOP Key Idea



AspeCt-oriented C Overview



History & Feature

- **Designed by Michael Gong and Hans-Arno Jacobsen**
 - started around May/June 2006, <http://www.aspectc.net>
- **ACC is an aspect-oriented extension to C**
- **Features**
 - most well known AOP features:
 - call and execution join points, before/after/around advice
 - composite pointcut, named pointcut
 - wildcard matching
 - cflow(), proceed(),
 - thisJoinPoint
 - add new member to existing struct/union
 - ACC-specific features:
 - **callp() join point**
 - **try-catch-throw exception handling**
 - **prereturn() inside advice**

Case Studies

ACC v0.6

- GNU libavl v 2.0.2a www.aspectc.net
 - *red-black tree (RBT) routines*
 - 4 aspects → 16 instances of RBT
- Nut/OS
 - a simple real-time OS in Ethernut project
 - modularize tracing functionality by aspect
- GCC 4
 - add logging functionality to GCC
 - output every internal function calls during compilation