AspectC++

Bringing Aspects into Deeply Embedded Systems

University of Erlangen-Nuremberg
Computer Science 4

Overview

- AspectC++ Project
- Language features
- Embedded weather station
- Conclusions

The AspectC++ Project

- Started in 2001
- Aim:
  - providing a feature-rich general purpose aspect language based on C++
- Inspired by AspectJ:
  - Similar language concepts and constructs
  - Extensions, alternations, enhancements where needed

Availability

- Free version can be downloaded from www.aspectc.org
- AspectC++ 0.7pre2 is a binary release only
- Final 0.7 release will include C++ source code (Open Source License)
- Commercially supported versions available from www.pure-systems.com

Language Status

- New and enhanced C++ parser in 0.7pre2
  - Supports templates, exceptions, namespaces
  - Extensions for GNU, Visual C++,...
  - Still some minor parser problems to be fixed before final 0.7 release
- New keywords: *aspect, advice, pointcut*

Features

- Based on Source-Code Transformation
- Free choice of backend compiler (GCC, Visual C++, Borland C++, …)
- Generates code with minimal runtime overhead
- Suitable even for deeply embedded systems

Demo Motivation

If AspectC++ can be used to develop software even for an 8-bit CPU with 4kB RAM, you can really use it everywhere!

Demo Scenario

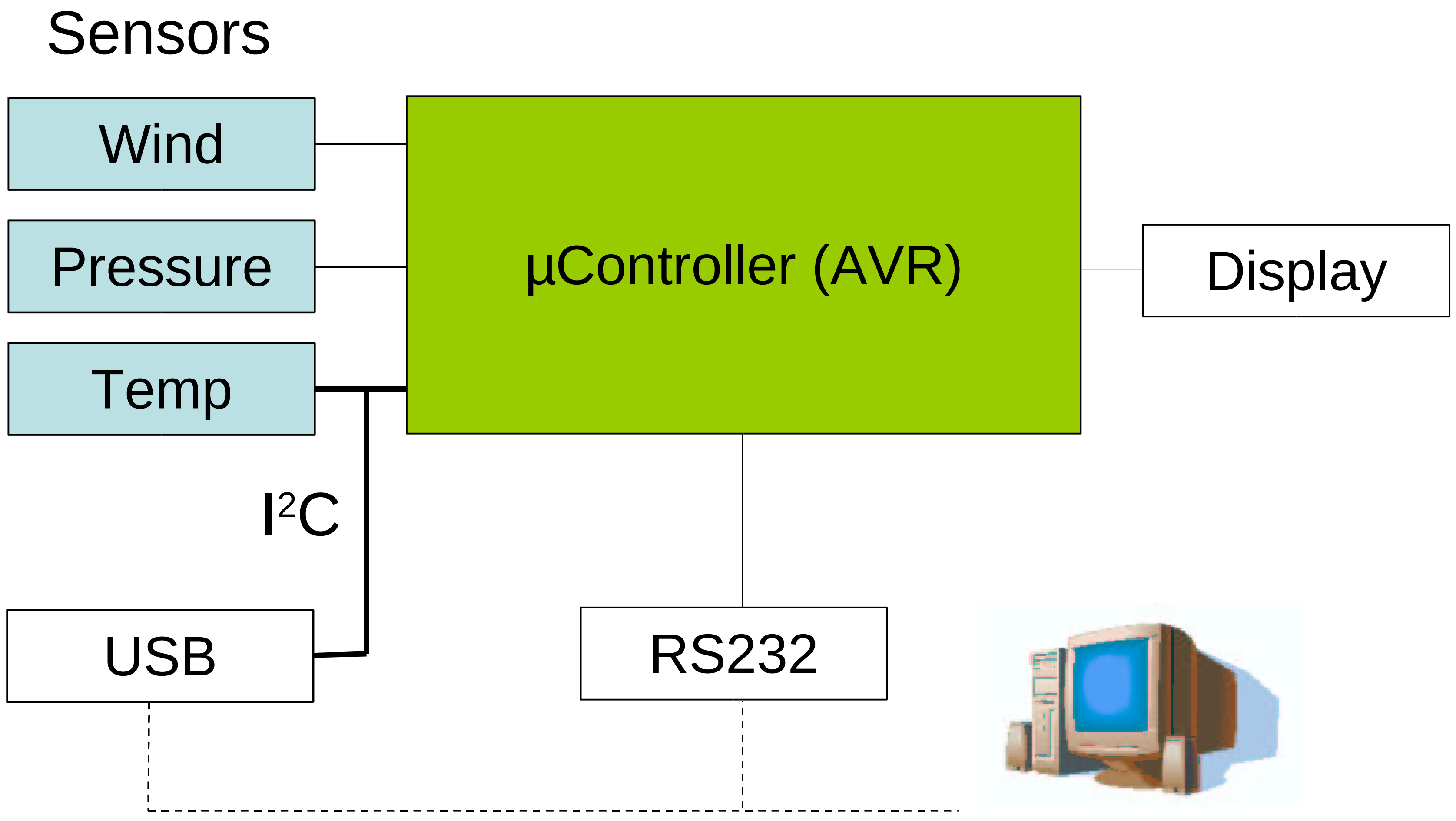
We will use AspectC++ to build weather station software for an embedded platform.

Target Platform

- 4 MHz 8-Bit AVR RISC CPU
- 4kB of RAM, 4kb of EEPROM
- Program stored in Flash Memory
- Remember:

  “Hello World” with Java needs 20 MB of RAM on a PC!

Demo Application

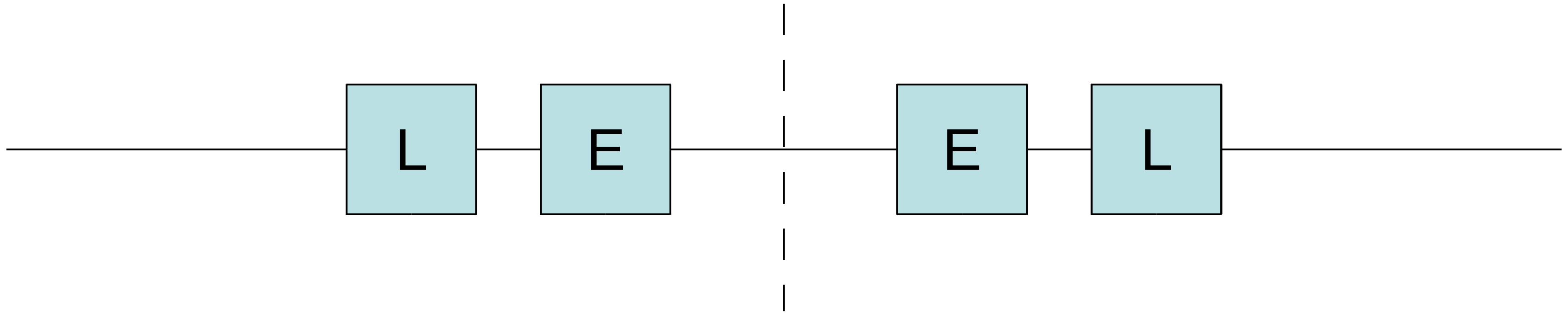


UDP/IP or simple protocol over USB/RS232

Join Point Model

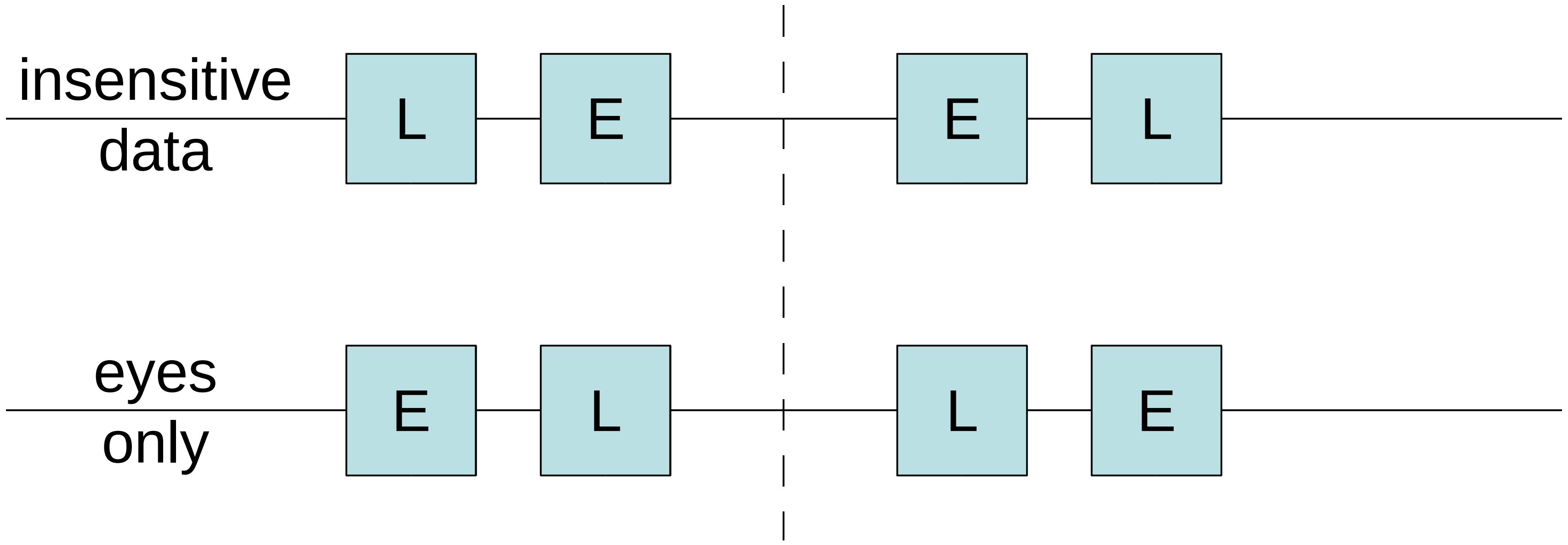
- Adopted from AspectJ
- Addition: *Name Join Points*
  - Correspond to GTNs in AspectJ
  - Uniform treatment of Name Join Points and Code Join Points in AspectC++
- Name Join Points can be part of virtual pointcut expressions

Aspect Ordering



- Consider dataflow logging for debugging purposes in a transaction system
- Data is always logged before encrypted
- AspectJ: `Log dominates Encrypt`

Aspect Ordering (2)



- Within same application
  - Log insensitive data in clear text to ease debugging
  - Use encryption for sensitive data

Aspect Ordering (3)

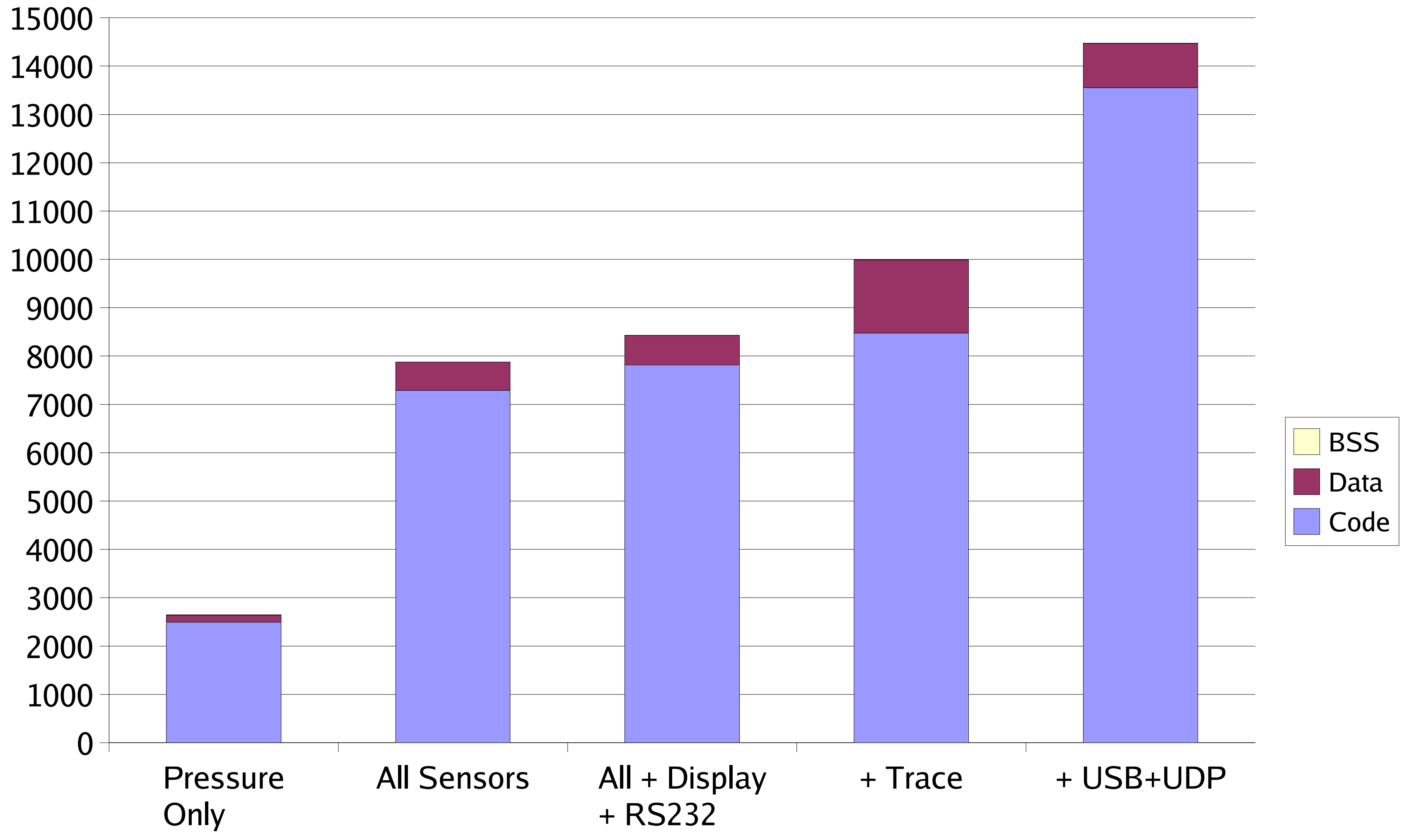
- In AspectC++:

```
aspect Encrypt { … };
aspect Log { … };
aspect Ordering_LE_for_Sensitive_Data {
  advice within(“Confidential”) :
    order(Encrypt,Log); };
  advice !within(“Confidential”) :
    order(Log,Encrypt); };
};
```

Aspect Ordering (4)

- Aspect ordering can be selected for specific join points
- Different subsystem can be treated differently
- Aspect ordering is separated out from the aspect declaration itself
  - Especially useful for reusable aspects

Code Sizes


15000
14000
13000
12000
11000
10000
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
Pressure Only
All Sensors
All + Display + RS232
+ Trace
+ USB+UDP
BSS
Data
Code

Conclusions

- AspectC++ is a feature-rich, simple to use, general purpose aspect language
- Distributed free of charge for non-commercial purposes, commercial licenses available
- Efficiency of C++ makes AOP available for a number of new target domains

Future Plans

- Further C++ parser enhancements
- More complete documentation, more example code
- Source code release, migration to an open source development model

http://www.aspectc.org