Developing Embedded Software Product Lines with AspectC++ *

Daniel Lohmann Olaf Spinczyk
Friedrich-Alexander-University Erlangen-Nuremberg
{lohmann,spinczyk}@cs.fau.de

Abstract

AspectC++ is a general purpose aspect-oriented language extension to C++. It is aimed to bring fully-fledged aspect-oriented programming (AOP) support in areas with strong demands on runtime efficiency and code density. This makes it possible to exploit the power of AOP for the domain of (deeply) embedded systems, where computation speed and available memory resources are strictly limited. AOP concepts are particularly useful for the development of scalable embedded system product lines. This will be demonstrated by a real world example: a small embedded device equipped with meteorological sensors and an 8-bit micro-controller running AspectC++ code. By covering the complete build cycle of configuration, compilation and installation, participants will understand, how easy it is to integrate AspectC++ with an existing tool chain. A presentation of the AspectC++ tools for Eclipse and the pure::variants variant-management system rounds up the demonstration.

Categories and Subject Descriptors D.2.2 [*Software Engineering*]: Design Tools and Techniques; D.3.3 [*Programming Languages*]: Language Constructs and Features

General Terms Languages, Measurement, Design

Keywords Aspect-oriented Programming (AOP), Software Prodcut Lines, AspectC++, Embedded Systems, Footprint

1. Introduction

AspectC++ is a general purpose aspect-oriented language extension to C++ designed by the authors and others[3]. It is aimed to support the well-known AspectJ programming style of AOP in areas with stronger demands on runtime efficiency and/or code density. One of these areas is the rapidly growing domain of embedded and deeply embedded systems. This domain is characterized by extreme constraints in memory and processing power (typically 8-bit micro-controllers and only a few kilobytes of RAM). AspectC++ enables developers to take advantage of AOP concepts for the development of embedded system product lines, as it combines fully-fledged AOP support with minimal overhead in the resulting code.

Besides providing a better encapsulation of typical cross-cutting concerns (such as synchronization, instrumentation, tracing...) AOP concepts are particularly useful for the development of highly flexible, yet resource-efficient, embedded software product lines. In this domain, AOP can even be superficial to object-orientation, as it provides excellent separation of concerns with practically no additional costs compared to a plain C approach with preprocessor-based configuration[2].

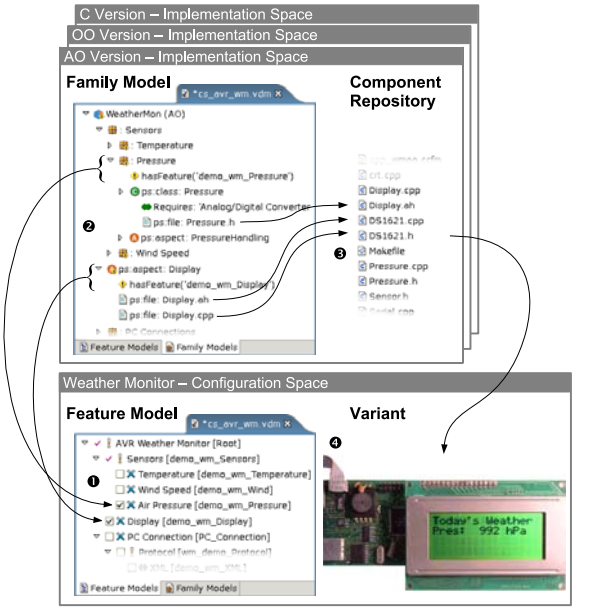

Figure 1. Weather station configuration process in Eclipse. In the *configuration space* the user selects all wished features from the *feature model* (1). The *family model* maps features to *logical implementation components* (2), which in turn are mapped to *physical implementation files* (3). The thereby determined set of implementation files for a concrete configuration is used to build the actual *weather station variant* (4).

2. The Demo

The goal of the demonstration is to show, that AOP is

applicable for the embedded systems domain in general and for the development of embedded product lines in particular.

affordable with respect to the resource consumption of the generated code.

usable today as the related tools have reached a high level of maturity.

The product line development process and the AspectC++ tool chain will be demonstrated “hands on” with a real world example.

* This work was partly supported by the German Research Council (DFG) under grant no. SCHR 603/4 and SP 968/2-1

OOPSLA'06, October 22–26, 2006, Portland, Oregon, USA.
ACM 1-59593-491-X/06/0010.

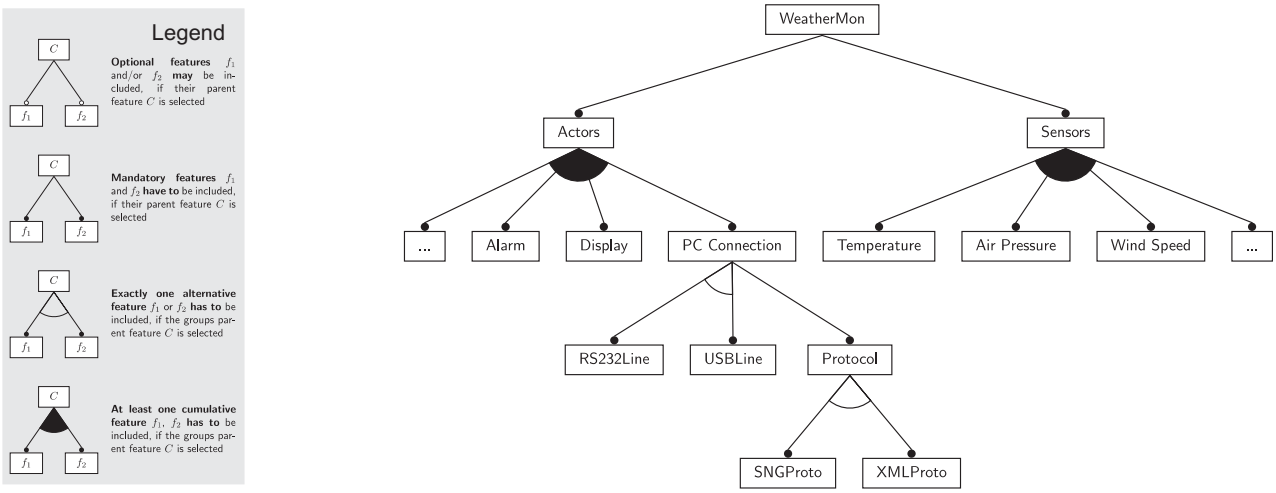

Figure 2. Feature diagram describing the available configurations of the embedded weather station product line.

The example is an embedded weather station product line. Participants will understand, how aspects can be used to implement product variants (Figure 2) that scale well in terms of provided features and required resources (Figure 3).

3. The Tools

A state-of-the-art Eclipse-based tool chain for embedded product line development with AOP (Figure 1) will be presented :

pure::variants: the industry-strength variant management and configuration system for Eclipse[1], developed and supported by pure::systems GmbH, Magdeburg, Germany. A free community edition is available under http://www.pure-systems.com.

ac++: the Open-Source weaver for the AspectC++ language[3] transforms AspectC++ code into C++ code. ac++ is easy to integrate into existing tool-chains and can be used in combination with any standard-compliant C++ compiler such as g++ and VisualC++. The AspectC++ documentation and tools are available under http://www.aspectc.org.

ACDT: the AspectC++ development tools for Eclipse provide an easy and sophisticated integration of AspectC++ into the Eclipse IDE. By features such as *join-point visualization*, *debugger integration*, and extra *crosscutting views*, developers have always a complete overview on how aspects interact with other parts of the code. The ACDT are Open-Source and available under http://acdt.aspect.org.

4. The Presenters

Daniel worked as software developer, consultant and trainer for several years. He finished his Diploma in Computer Science in 2002 as best student of grade. His PhD research is on the development of aspect-oriented operating system product-lines. Since joining the Operating Systems group at Friedrich-Alexander-University, he actively participates in the AspectC++ language design and the ac++ development. His main focus is the application of aspects to embedded system software.

Olaf has a background of more than eight years research on AOP and system software product-lines. In 2002 he got the "best dissertation of 2002" award by the computer science faculty of the University of Magdeburg, Germany, for his work in this field. Today he is the main designer and developer of the ac++ weaver. In 2002 he started to cooperate with pure-systems GmbH in Magdeburg, Germany, to speed up the ac++ development and to evolve it from a research prototype to a commercial product.

References

[1] Danilo Beuche. Variant management with pure::variants. Technical report, pure-systems GmbH, 2003. http://www.pure-systems.com/.

[2] Daniel Lohmann, Olaf Spinczyk, and Wolfgang Schröder-Preikschat. Lean and efficient system software product lines: Where aspects beat objects. In *Transactions on AOSD, Special Issue on Aspect-Oriented Programming for Systems Software and Middleware*, LNCS. Springer, 2006. (to appear).

[3] Olaf Spinczyk, Daniel Lohmann, and Matthias Urban. The design and implementation of AspectC++. In *Journal on Knowledge-Based Systems, Special Issue on Creative Software Design*. Elsevier, 2006. (to appear).

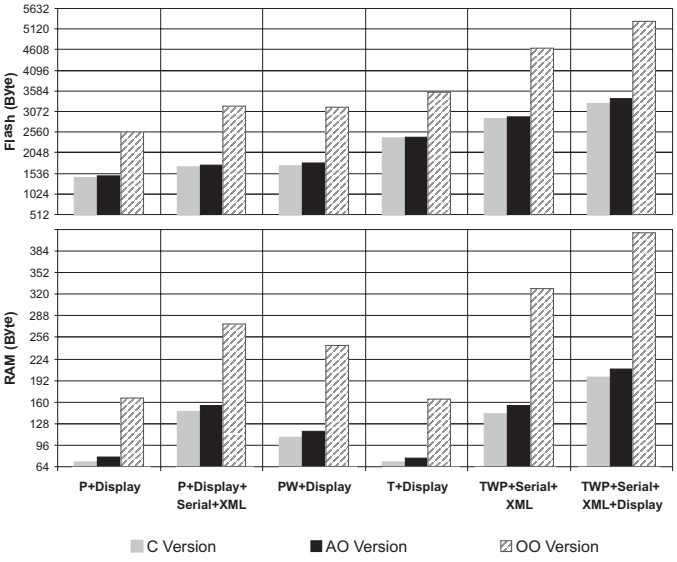

Figure 3. Footprint comparison (RAM, flash) for different weather station configurations