Introduction to the Attribute Driven Design Method[1]

Felix Bachmann
Carnegie Bosch Institute
Carnegie Mellon University
Pittsburgh, Pa 15213 USA
+1 412 268 6194
fb@sei.cmu.edu

Len Bass
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, Pa 15213 USA
+1 412 268 6763
ljb@sei.cmu.edu

ABSTRACT
This tutorial will introduce the Attribute Driven Design (ADD) method. ADD is a method for designing the software architecture of a system or collection of systems based on an explicit articulation of the quality attribute goals for the system(s). The method is appropriate for any quality attributes but has been particularly elaborated for the attributes of performance, modifiability, security, reliability/availability and usability. The method has been used for designing the software architecture of products ranging from embedded to information systems.

Keywords
Software architecture design, software architecture, software design methods

1 INTRODUCTION

It has long been recognized that quality attributes are, in large part, determined by the software architecture of a system. This recognition has been the basis for several software architecture analysis methods[4,5]. The application of this recognition to the problem of designing the software architecture is a recent development, however. The ADD[2] [1] method is based on understanding the relationship between software qualities and the architecture mechanisms used to achieve these qualities.

The ADD method is designed to achieve two major goals: supporting the earliest stages of the design process where the ability to achieve desired quality attributes is determined and allowing design to begin early enough in the life cycle to support modern time-to-market requirements. We now discuss these two goals.

The conceptual architecture

The ADD method is narrowly focused on the initial design stages of the life cycle. The term "conceptual architecture" [3] has been coined to describe this initial design stage. ADD is a method for designing the conceptual architecture. The conceptual architecture encompasses the first set of design decisions during a design process. The conceptual architecture includes the decomposition of function for the final system, identification of possible threads of parallelism and identification of possible physical network configurations and the allocation of the functional units to processors. Because the basis for the quality attributes of the designed system are established when the initial decisions are made and because the conceptual architecture encompasses the initial set of design decisions, focussing on the conceptual architecture (as the ADD Method does) is essential to achieve desired quality attributes.

Furthermore, because the ADD Method is narrowly scoped on the initial portion of the architecture design process, it is compatible with methods that focus on other portions of the life cycle such as requirements gathering or on detailed design. It has been used in conjunction with several different tool sets. Since the method is based on an explicit articulation of quality attribute goals, it does assume that the requirements gathering process captures both abstract and concrete expressions of quality attribute requirements.

Allowing for uncertainty in requirements

Getting the requirements complete and correct is a time consuming task. Some would argue that it is also never ending because requirements change as the customer achieves greater understanding of the system being designed and as circumstances change. In any case, in order to satisfy modern time-to-market constraints, a design method has to be able to deal with uncertainty in requirements.

The ADD method depends on a determination of "architectural drivers". The architectural drivers for a design are those combination of functional, quality and business requirements that give a shape to the architecture. Design using the ADD method can begin once those

1. This work supported by the U.S. Department of Defense and Robert Bosch, GmbH.

2. This method was previously called the "Architecture Based Design" Method but ABD is a trademark of Lockheed Martin, Inc.

0-7695-1050-7/01 $10.00

drivers have been determined to a reasonable degree of certainty. This allows the refinement of requirements and design to be done in parallel and allows ADD to be used in a short time-to-market situation.

2 ARCHITECTURE MECHANISMS AND QUALITY ATTRIBUTES

The ADD method is based on understanding how quality attributes are achieved through architecture mechanisms. The mechanisms to achieve quality attributes are known by experienced architects but only recently has an effort begun to codify these mechanisms [2]. The architectural drivers lead, through the use of architecture mechanisms, to a design that will satisfy the quality requirements for a system.

One step of the method involves identifying the architecture mechanisms that are available to satisfy the particular architectural drivers. This identification depends both on what has been codified and the knowledge of the architect. Currently, the identification of the architecture mechanisms relies heavily on the knowledge of the architect but the goal is to support the architect with an enumeration of mechanisms and the impact of these mechanisms on all of the quality attributes.

3 THE ADD METHOD

The method itself is a recursive decomposition method. The overall system is decomposed into a collection of children called "conceptual subsystems" and these, in turn, are decomposed into "conceptual components." Because the method is recursive, the same steps apply whether they are being applied to decomposing the system or the conceptual subsystems. In theory, the method could also be applied to decompose the conceptual components but, thus far, conceptual components have been adequate to capture the conceptual architecture.

At each major step of the method, the architectural drivers for the element being decomposed are identified, the element is decomposed and the decomposition is verified according to the quality attribute requirements and use cases representing the essential functionality. The constraints are also checked at each major step.

Views and quality attributes

Three views are used in the ADD method. They are the module view, the component and connector view and the deployment view. The module view is used to capture the responsibilities of each element in the decomposition including infrastructure responsibilities such as resource management. The module view is also used to capture the information flow among elements and the information aspects of the interfaces of the elements.

The component and connector view is used to reason about dynamic aspects of the system such as instances of components, contention for resources and synchronization points. The deployment view is used to reason about allocation to physical hardware.

The achievement of each quality of concern depends, in part, on information in each of the different views. Modifiability, for instance, depends on the allocation of functionality to the elements and this is reflected in the module view. It may also depend on allocation to hardware and this is reflected in the deployment view. Performance, for another example, depends on the resources required to satisfy responsibilities (the module view), synchronization details (the component and connector view) and allocation to various processors (the deployment view). The same is true for the other attributes.

4 CONCLUSION

The ADD method has been used by various groups within Robert Bosch, GmbH to define the conceptual architecture for products that are intended to provide the basis for product lines. Some of these defined products are now approaching production. The Bosch groups have been able to internalize and apply the method with little training since the method is narrowly scoped and is quite intuitive for experienced designers.

5 REFERENCES

1. Bachmann F., Bass L., Chastek G., Donohoe P., Peruzzi F., "The Architecture Based Design Method" CMU/SEI-2000-TR-001, Carnegie Mellon, Pittsburgh, January 2000.
2. Bass, L., Klein, M., Bachmann, F. "Quality Attribute Design Primitives" CMU/SEI-2000-TN-017, Carnegie Mellon, Pittsburgh, December, 2000
3. Hofmeister C., Nord R, Soni D., "Applied Software Architecture" Addison Wesley Publish, Reading, Ma, 2000.
4. Kazman R., Abowd, G., Bass L, and Clements P., "Scenario Based Analysis of Software Architecture", IEEE Software, November 1996 pp 257-266.
5. Kazman, R.; Barbacci, M.; Klein, M.; Carriere, S. J.; & Woods, S. G. "Experience with Performing Architecture Tradeoff Analysis,"54-64. Proceedings of the 21st International Conference on Software Engineering (ICSE 21). Los Angeles, CA, May 1999.