Software Architecture Modelling, Analysis and Implementation with SoftArch

John Grundy
Department of Computer Science, University of Auckland
Private Bag 92019, Auckland, New Zealand
john-g@cs.auckland.ac.nz

Abstract

Good software architecture design is crucial in successfully realising an OOA specification with an appropriate OOD model that meets the specification’s functional and non-functional requirements. Unfortunately most CASE tools and software architecture design notations do not adequately support software architecture modelling and analysis, nor integration with OOA & D models. We describe SoftArch, an environment which provides flexible software architecture modelling using a concept of successive refinement. SoftArch also provides extensible analysis tools enabling developers to analyse their architecture model properties. This paper overviews the motivation for SoftArch, its modelling and analysis capabilities, and its integration with various analysis, design and implementation tools.

1. Introduction

Many software modelling notations and tools have been developed [6, 7, 12, 15], and there has been an increasing emphasis on software architecture modelling in addition to OOA & D modelling in CASE tools. Various approaches have been tried, including those of UML [3], PARSE [15], JViews and aspects [8, 10], tool abstraction [7], and Clock [6, 22]. Support tools include Rational Rose [18], JComposer [10], PARSE-DAT [15], ViTABaL [7], SAAMTool [12] and Argo/UML [19].

Most of these systems provide partial software architecture modelling solutions, with only some aspects of architecture modelling supported e.g. basic structure, limited dynamic behaviour and event models, dynamic process creation etc [11, 14]. Few provide adequate analysis tools to help developers reason about their models and ensure OOA requirements are met and all software architecture components are refined to suitable OOD abstractions [19, 11]. Few support OOD and/or implementation code generation from architecture-level abstractions, and few support reuse of previously developed models and patterns [22, 19]. Almost none allow new architecture abstractions and analysis tools to be added, and most have poor or no integration with OOA, design and implementation tools.

We describe SoftArch, a new, extensible environment using new approaches to software architecture modelling, analysis, design generation and tool integration. SoftArch uses an extensible meta-model of architecture abstractions. Architects use a flexible, extensible visual notation based on allowable abstractions to describe and refine software architecture models, including links to OOA and design objects and classes as appropriate. A collection of extensible “analysis agents” constrain, guide and advise architects as they build and refine their architecture models. SoftArch has been integrated with several OOA, design and implementation tools, as well as a process management environment, using a variety of tool integration techniques.

The following section presents a motivation for our work developing SoftArch. We then overview the environment’s capabilities, and in the following sections describe its software architecture modelling, refinement and analysis support. A brief discussion of SoftArch’s implementation and architecture is presented, and we conclude with a summary of the contributions of this work and directions for future research.

2. Motivation

Software architecture development has become an increasingly important part of the software lifecycle, due to the increasing complexity of software being constructed [1, 3, 21]. Software developers need to carefully describe and reason about the architectures of complex, distributed information systems, which are often comprised of a mix of new and reused components. A good, extensible and maintainable architecture often makes the difference between successful and failed projects. Much more time tends to be spent on architecture development than previously, and many more options exist for developers [1].