ArchStudio 4: An Architecture-Based Meta-modeling Environment

Eric Dashofy, Hazel Asuncion, Scott Hendrickson, Girish Suryanarayana, John Georgas, Richard Taylor
Department of Informatics, University of California, Irvine
{edashofy, hasuncio, shendric, sgirish, jgeorgas, taylor}@ics.uci.edu

Abstract

We will demonstrate ArchStudio, an environment for software architecture modeling and meta-modeling. We will also showcase a set of innovative architecture-centric applications that use ArchStudio technologies as their basis.[1]

1. Introduction

A recurring question in the world of software architecture modeling is how to support the diverse array of issues, concerns, and stakeholders that occur in complex software systems. Early architecture description languages (ADLs) [5] were limited to component-and-connector modeling, with additional distinguishing features. Later ADLs, including UML 2.0 [1], AADL [3], Koala [6], and xADL 2.0 [2] are distinguished by a broader perspective on architecture. The concerns addressed by these ADLs go beyond the narrow technological foci of the first-generation ADLs, taking into account domain-specific and business concerns.

To address this question, we have developed a software architecture-based development environment called ArchStudio. ArchStudio addresses the above challenge by providing well-defined extensibility mechanisms for all of its capabilities: architecture modeling, visualization, analysis, and tool integration. This makes it an ideal platform for investigating and comparing new architectural approaches and research directions. In addition to providing a basis for investigating new features, ArchStudio supports a distinctive set of architecture capabilities, including hierarchical modeling, product-line modeling, and dynamic evolution. ArchStudio and its constituent technologies have supported more than a dozen published projects from academia and industry.

The latest release, ArchStudio 4, is an Eclipse-based environment that runs on Windows, UNIX/Linux, MacOS, and other platforms that support Eclipse and Java. ArchStudio is an open-source project and released under a BSD-style license, encouraging both academic and industrial adoption.

We describe ArchStudio's capabilities, focusing on the extensible aspects of the environment (modeling, visualization, and analysis). We conclude with descriptions of innovative projects that leverage ArchStudio.

2. ArchStudio

The primary modeling notation used by ArchStudio is xADL 2.0, an extensible, XML-based architecture description language [2]. xADL 2.0 is *modular*, rather than monolithic. Modeling features in xADL 2.0—for example, the ability to model components and connectors, or the ability to model points of variability in an architecture—are broken up into modules. Each module is defined in an XML schema. To date, we have developed modules for capturing traditional concerns like components and connectors along with more innovative concerns such as product-lines, implementation mappings, architectural states, and so on. Tools read the xADL 2.0 schemas and automatically generate a data binding library that provides a foundation for other tools. Thus, users can extend the xADL language with new features and automatically generate libraries used for building tools that interact with those new features.

ArchStudio includes two primary editors that support xADL 2.0 extensions. The first is *ArchEdit*, a tree- and table-based editor. ArchEdit is syntax-directed, so the contents of its user interface are generated automatically based on the underlying xADL schemas. A second editor, called *Archipelago*, provides a symbolic point-and-click boxes-and-arrows style editing interface. All aspects of the editor are constructed using a plug-in based framework, allowing plug-in 'packages' to be added to support new xADL schemas.

ArchStudio includes a plug-in-based framework for integrating analysis tests called *ArchLight*. ArchLight provides a unified user interface in which various tests

[1] Work supported by NSF Grants CCF-0430066 & 0524033, CCR-0093489, DUE-0536203, IIS-0205724 and the Boeing Company.

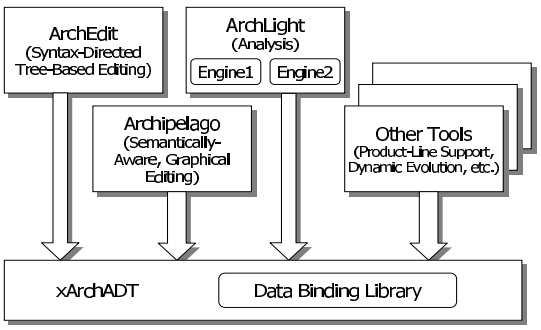

Figure 1. Major ArchStudio tools interacting through xArchADT.

can be applied to and run on architectural models. ArchStudio includes and adapts the Schematron XML validator for use as an architectural analysis engine. Schematron allows tests to be defined as compact constraints over the XML structure of xADL documents. As in Archipelago, xADL schemas are associated with packages of ArchLight tests. A new xADL schema can be deployed along with tests to verify the correctness of the parts of the model specified in the schema.

ArchStudio is built using its own tools and technologies. The architecture of the ArchStudio environment is specified in xADL 2.0, including extensions that map its various components and connectors to Java implementations. This description is visualized in Archipelago and validated in ArchLight. When ArchStudio is started, a bootstrapping component instantiates the environment from the ArchStudio architecture description using the data binding library. Figure 1 shows the relationships between the primary ArchStudio tools.

3. Applications

Working with an industrial partner, we have applied ArchStudio to modeling the hardware/software architectures of software-defined radios (SDRs). SDRs are devices that exchange data via a radio link, where the radio is contolled by flexible, reconfigurable software. We applied ArchStudio in two contexts. First, we constructed a small translator program using ArchStudio's data binding library to translate SDR deployment specifications into xADL documents. Using Archipelago and ArchLight, we provided the SDR engineers with heretofore unavailable visualizations, as well as a list of about 800 concrete issues with the specifications.

Second, we extended ArchStudio to support the association of statecharts with product-line variants. Using existing UML component diagrams and sequence diagrams as a guide, we developed a model of the startup sequence of the radio's operating environment. The result was a sequence of Archipelago diagrams showing exactly which components were loaded at various points in the bootup sequence; an artifact previously unavailable to the SDR engineers.

EASEL [4] is a layer-based design environment built atop ArchStudio. EASEL explicitly and separately captures alternative design choices as *change sets* and composes them together, much like layers in layer-based graphical design tools, allowing a designer to incrementally explore designs consisting of different ideas and approaches. It also explicitly captures knowledge about design choice interactions as *relationships*, much like feature interactions are captured in a feature model, allowing a designer to explore only desired design choice combinations.

PACE is an architectural style for trust management in decentralized applications [7]. PACE provides specific guidance on how to incorporate trust models within an entity's architecture such that the resultant architecture is made secure against potential malicious attacks that are characteristic of open decentralized systems. PACE has been evaluated in the context of several domains including decentralized auctioning, decentralized file-sharing, and crisis response. In these domains, each PACE-based entity has been designed and built using the ArchStudio suite of tools.

4. References

[1] Booch, G., Rumbaugh, J., and Jacobson, I. The Unified Modeling Language User Guide. 2nd ed. Addison-Wesley Object Technology Series. Reading, Massachusetts: Addison-Wesley Professional, 2005.

[2] Dashofy, E., Hoek, A.v.d., and Taylor, R.N. A Comprehensive Approach for the Development of Modular Software Architecture Description Languages. ACM Transactions on Software Engineering and Methodology (TOSEM). 14(2), p. 199-245, April, 2005.

[3] Feiler, P.H., Lewis, B., and Vestal, S. The SAE Avionics Architecture Description Language Standard: A Basis for Model-Based Architecture-Driven Embedded Systems Engineering. In *Proc. of the RTAS 2003 Workshop on Model-Driven Embedded Systems*. Washington, D.C., 2003.

[4] Institute for Software Research. EASEL, An Extensible Architecting Support Environment with Layers. <http://www.isr.uci.edu/projects/easel/>.

[5] Medvidovic, N. and Taylor, R.N. A Classification and Comparison Framework for Software Architecture Description Languages. IEEE Transactions on Software Engineering. 26(1), p. 70-93, Jan., 2000.

[6] Ommering, R.v., Linden, F.v.d., Kramer, J., and Magee, J. The Koala Component Model for Consumer Electronics Software. IEEE Computer. 33(3), 78-85, Mar., 2000.

[7] Suryanarayana, G., Erenkrantz, J.R., and Taylor, R.N. An Architectural Approach for Decentralized Trust Management. IEEE Internet Comp. 9(6), p. 16-23, Nov/Dec, 2005.