Trinity: A Language for Multi-View Architecture Description and Control

Jonathan Aldrich
Carnegie Mellon University
jonathan.aldrich@cs.cmu.edu

Madaline Kirwin
Grinnell College
kirwinma@grinnell.edu

Selva Samuel
Carnegie Mellon University
ssamuel@cs.cmu.edu

Abstract

This is the text of the abstract.

CCS Concepts • **Software and its engineering** → **General programming languages**; • **Theory of computation** → *Program analysis*

Keywords keyword1, keyword2

1. Multiple Views of Software Architecture

– Give the SEI definition of software architecture

– State the usefulness of software architecture in analysis and design activities

– Introduce ADLs.

– Introduce views. State their usefulness in dealing with complexity by separating concerns

– State the lack of support for multiple views in ADLs.

– State that industry practitioners desire support for multiple views. Cite study by Malavolta et. al.

The software architecture of a system is the set of structures needed to reason about the system, which comprise software elements, relations among them, and properties of both []. [1]

2. Architectural Control

– What is architectural control? –> Architectural constraints must be enforced in the implementation

– Introduce communication integrity

– Describe ArchJava's support for enforcing communication integrity in a single JVM and its failure to do so across multiple JVMs

3. Trinity

Acknowledgments

Acknowledgments, if needed.

References

[1] I. Malavolta, P. Lago, H. Muccini, P. Pelliccione, and A. Tang. What Industry Needs from Architectural Languages: A Survey. *IEEE Transactions on Software Engineering*, 39(6):869–891, June 2013.