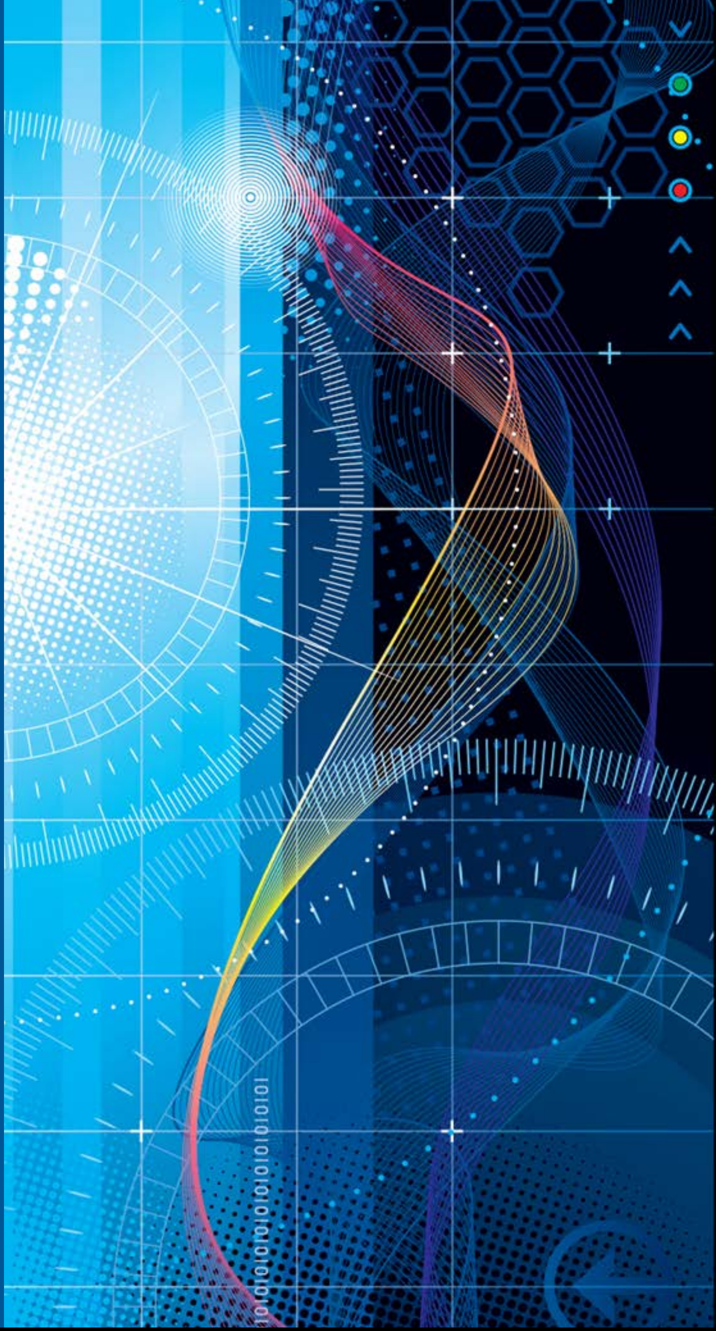
AADL to AltaRica Translation

Lutz Wrage

October 2016

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213



This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN “AS-IS” BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

DM-0004075

Goal

Convert models in AADL to AltaRica

- Core AADL determines the hierarchical structure of the system
- Error Model Annex determines
  - Fault propagation between components
  - Internal fault behavior of components

Determine the correct mapping from (a subset of) AADL to AltaRica constructs

Implement an automated conversion as add-on to the OSATE (Open Source AADL Tool Environment)

Create a library of small example models to

- Illustrate the transformation
- Serve as a set of test cases for the automated transformation

Progress: Mapping Definition$_1$

At a high level

AADL Construct	AltaRica Construct(s)
Component	Class/Node
Error propagation and connection / binding	Flow variables and external assertion (between flows of sub-nodes)
Component error behavior	Node internal states, transitions, assertions
Error states	One state variable
Error events	Events
Error transitions	Additional events, transitions
Out propagations	Assertions
Event occurrence properties	Delay attributes (on events)
Modes	Select system operation mode before transformation

Progress: Mapping Definition$_2$

Challenges

- AADL supports fan-in for error propagations; AltaRica does not:
  - Error propagation points must be split in AltaRica to allow unique definition of types (domains) for AltaRica flow variables
- AADL error types are organized in a generalization hierarchy:
  - Error types need to be expanded to leaf types
- Error propagations and states in AADL use sets of error types; AltaRica does not support sets:
  - Matching of subsets must be translated to Boolean expressions
  - Need to expand typed error states according to all possible subsets
- AADL supports error type products to model simultaneous occurrence of errors; AltaRica does not:
  - Need to translate a type product into a single domain constant
- How to treat AADL modes?
  - Select one system operation mode and translate AADL model for that mode

Progress: Automated Translation$_1$

Challenge

- No re-usable open source implementation of AltaRica 2.1 available:
  - LaBRI tools use outdated AltaRica version, not Eclipse-based
  - OpenAltaRica AR3 editor lacks needed functionality, e.g., name resolution, and there are licensing issues
- → Implemented support for subset of AltaRica in OSATE

Implemented transformation of

- AADL components to AltaRica classes
  - An AADL instance model defines the scope of the transformation
  - Components with error behavior define the depth of the transformed structure (AADL components at a deeper nesting level don’t contribute to error behavior)
- Error events to AltaRica events
- Error propagations to flow variables (including fan-in handling)
- Error propagation path via connections and bindings to assertions about flow variables in different classes
- Error behavior transitions to AltaRica transitions and assertions inside a class
- Occurrence properties to AltaRica event attributes

Progress: Automated Translation$_2$

Implemented in Java and ATL (Atlas Transformation Language)

Fully integrated into OSATE

Some non-essential AADL error model features are not supported by the translator

- Support for the most common subset of error transition condition and out propagation condition expressions
- Support for un-typed AADL error states only
- No support for branching (non-deterministic) transitions
- No support type transformations and type mappings

Progress: Example Library

A set of example models has been created

- Starting point: a small library of AltaRica patterns
- Created corresponding AADL models
- Tested translation back to AltaRica
- Resulting AltaRica models can be analyzed in OpenAltaRica
- Created demonstration video using the triple-redundant voter example