The Montana Toolset: Formal Analysis of AADL Specifications

SAE AS-2 Working Group
Seal Beach, California
27 January 2005
Outline

► Origins, Goals, Plans and Strategy
► ACSR and the VERSA Toolkit
► The Charon Language and Toolkit
► The Montana Toolkit
  - STTR Phase I
  - STTR Phase II
Origins

STTR AF04-T023

- Modeling Languages and Analysis Tools for Complex Distributed Systems
- **OBJECTIVE:** Develop language and computational tool support for modeling and analyzing complex distributed system designs and integrate these methods to build distributed systems.
Origins

►► AADL
- Language, Eclipse tools, OSATE, EMF

►► ACSR: Algebra of Communicating Shared Resources
- Real-Time
- Resources
- VERSA tool support

►► Charon
- Hybrid state machines
- Charon tool support
Goals, Plans and Strategy

► STTR Phase I
  ▪ Proof of Concept
    ► Translation; Annex
    ► Prototype
    ► Small case studies for demo

► STTR Phase II
  ▪ Primary R&D
    ► Tools Design and Development
    ► Methodology
    ► Proselytize

► Phase III
  ▪ Make the world a better place
Goals Plans and Strategy

► Phase I

- Eclipse, Osate, EMF
- Custom developed interface layer plug-ins
- C++ Implementation of VERSA
- Charon
Goals Plans and Strategy

Phase II

- Eclipse, ? (AADL front-end), EMF
- Custom developed analysis plug-ins
- Fresh implementation of VERSA
- Interface to special purpose analysis engines
  - CADP, Spin
  - PVS
- Books, tutorials, advocacy to support methodology
Outline

► Origins, Goals, Plans and Strategy
► ACSR and the VERSA Toolkit
► The Charon Language and Toolkit
► The Montana Toolkit
  - STTR Phase I
  - STTR Phase II
ACSR and the VERSA Toolkit

►► ACSR: Algebra of Communicating Shared Resources
  ▪ Process Algebra
  ▪ Real-time
    ▪ Timed Actions
    ▪ Timeouts
    ▪ Interrupts
  ▪ Resources
  ▪ Priorities
ACS R and the VERSA Toolkit

►► Timed Actions: \{((r1,p1),(r2,p2),\ldots)\}:P
►► Choice: \(P1 + P2\)
►► Parallel Composition: \(P1 || P2\)
►► Temporal Scope: 

\[ P \Delta^t_a (P_e, P_t, P_i) \]
►► Miscellaneous operators
ACSR and the VERSA Toolkit

\[ \text{TBB} = (\text{in,1}).\text{TBB1}; \]
\[ \text{TBB1} = (\text{in,1}).\text{TBB2} + (\text{out,2}).\text{TBB}; \]
\[ \text{TBB2} = (\text{out,2}).\text{TBB1}; \]
ACSR and the VERSA Toolkit

SYS = (OBBL|OBBR){sync};
OBBL = (in,1).(sync,3).OBBL;
OBBR = ('sync,3).(out,2).OBBR;
ACSR and the VERSA Toolkit

\[\text{OBBL} \xrightleftharpoons{(\text{in},1)} \text{OBBL'} \quad | \quad \text{OBBR} \xrightleftharpoons{('\text{sync},3)} \text{OBBR'}\]
ACSR and the VERSA Toolkit
ACSR and the VERSA Toolkit
ACSR and the VERSA Toolkit

► Utility
- State space exploration
- Equivalence testing

► VERSA: Verification Execution and Rewrite System for ACSR
- Syntax checking
- State space construction
- State space exploration
- Equivalence testing
- Term rewriting
ACSR and the VERSA Toolkit

Utility

- Traditional Schedulability Analysis
- Schedulability Analysis for Arbitrary Systems
Translating AADL to ACSR

- Threads are modeled as ACSR processes
  - Based on thread semantic automaton
- Processors and access connections are modeled as resources
- Event and data connections are modeled as communication channels
Example: Cruise Control

► Standard example (from OSATE release)

+ auxiliary processes for bookkeeping
Example: Cruise Control

- Processor and connection bindings determine resources
- Scheduling protocol determines priorities
- Periodic processes have activators
ACSR and the VERSA Toolkit

- Eclipse + OSATE + EMF + Translator + VERSA + reinterpretation of results in AADL
  = Schedulability analysis for threads
Outline

- Origins, Goals, Plans and Strategy
- ACSR and the VERSA Toolkit
- The Charon Language and Toolkit
- The Montana Toolkit
  - STTR Phase I
  - STTR Phase II
Hybrid Automata: Formalism for Hybrid System Models

- Continuous dynamics: Mathematical equation
  - Differential equation
    - $x' = 1 \ (dx/dt = 1)$: constant increase
    - $x' = x \ (dx/dt = x)$: exponential increase ($x = e^t$)
  - Algebraic equation
    - $y = \sin(x)$
  - Invariant
    - $x \geq -10$

- Discrete control: Finite State Machine
  - State: dynamics
    - $x' = 1, \ x' = -1, \ x' = x, \ x' = -x, \ ...$
  - Transition: switching of dynamics
    - $x' = 1 \ (x > 10) \rightarrow x' = -1$
CHARON Language Features

- Individual components described as agents
  - Composition, instantiation, and hiding

- Individual behaviors described as modes
  - Encapsulation, instantiation, and scoping

- Support for concurrency
  - Shared variables as well as message passing

- Support for discrete and continuous behavior
  - Differential as well as algebraic constraints
  - Discrete transitions can call Java routines
Modes and Agents

- Modes describe sequential behavior
- Agents describe concurrency

Syntax:

- {t = 1} • local t, rate, global level, infusion
- {level ∈ [2, 10]} • level ∈ [4, 8]
- Compute
t=10
e
dx de
t::=0
- Maintain
{t<10}
- Normal

Agent Controller

Emergency

- dx de
- level ∈ [2, 10]

Global level
- global infusion
- {level = f(infusion)}

Agent Tank
Charon toolset: visual editor
Charon toolset: visual editor
Charon toolset: control panel

```java
extern real java.lang.Math.random();

// a leaky tank controlled by a pump
agent LeakyTank()
    private analog real level, flow;
    agent pump = Pump(.5, 10) [ flow := flow ]
    agent tank = LTank() [ inflow := flow ]

// the tank agent with a hidden leak
agent LTank()
    private discrete real leak;
    agent tank = Tank();
    agent hole = Hole();

// the hole agent: leak changes randomly every so often
agent Hole()
    write discrete real leak;
    init { leak = 1; }
    mode top = HoleMode();

mode HoleMode()
    write discrete real leak;
```
Charon toolset: simulation
Charon toolset: simulation
Case Study
Four-legged Robot: Architectural Model

► Input
  ▪ touch sensors

► Output
  ▪ desired angles of each joint

► Components
  ▪ Brain: control four legs
  ▪ Four legs: control servo motors
  ▪ Instantiated from the same pattern
Case Study
Four-legged Robot: Behavioral Model

► Control objective
  - \( v = c \)

► High-level control laws
  - \( x = -v \) \( x \geq -\text{stride} / 2 \)
  - \( y = kv \)

► Low-level control laws
  - \( j_1 = \arctan(x / y) - \arccos(\frac{x^2 + y^2 + L_1^2 - L_2^2}{2L_1 \sqrt{x^2 + y^2}}) \)
  - \( j_2 = \arccos(\frac{x^2 + y^2 + L_1^2 - L_2^2}{2L_1 L_2}) \)
Code Generation Framework

- **Front-end**
  - Translate CHARON objects into modular C++ objects

- **Back-end**
  - Map C++ objects to execution environment

CHARON objects

- agent
- mode
- diff eqn
- transition
- analog var

C++ objects

- class agent
- class mode
  - diff()
  - trans()
- class var

Execution environment

- scheduler
- API

Target platform
Outline

- Origins, Goals, Plans and Strategy
- ACSR and the VERSA Toolkit
- The Charon Language and Toolkit
- The Montana Toolkit
  - STTR Phase I
  - STTR Phase II
The Montana Toolkit – Phase I

► Translation of restricted model to ACSR
  ▪ Schedulability analysis of threads
  ▪ General state space exploration
  ▪ Simulation

► Definition of Charon behavioral annex
  ▪ Simulation
  ▪ Generation of code from...
    ► AADL architecture
    ► Embedded Charon
The Montana Toolkit – Phase II

► Robust interpretation of AADL in ACSR
  - Model checking
  - Schedulability analysis
  - Equivalence preserving system refactoring
  - Equivalence testing
  - Simulation

► Complete behavioral annex based on Charon
  - Property checking
  - Simulation
  - Analysis
Outline

- Origins, Goals, Plans and Strategy
- ACSR and the VERSA Toolkit
- The Charon Language and Toolkit
- The Montana Toolkit
  - \textit{STTR Phase I}
  - \textit{STTR Phase II}
Contacts

► Duncan Clarke
  - [http://www.fremontassociates.com](http://www.fremontassociates.com)
  - [dclarke@fremontassociates.com](mailto:dclarke@fremontassociates.com)

► Oleg Sokolsky
  - [http://www.cis.upenn.edu/~sokolsky](http://www.cis.upenn.edu/~sokolsky)
  - [sokolsky@cis.upenn.edu](mailto:sokolsky@cis.upenn.edu)

► AFOSR STTR AF04-T023 Program Director:
  - Dr. Robert L. Herklotz
  - Program Manager: Software and Systems
  - Air Force Office of Scientific Research
  - 4015 Wilson Blvd., Room 713
  - Arlington, VA 22203-1954
  - email: robert.herklotz@afosr.af.mil
  - (703) 696-6565 fax (703) 696-8450