V&V MURI
Transition Strategy

Mani Chandy    John Doyle    Richard Murray
California Institute of Technology

Eric Klavins    Pablo Parrilo
U. Washington    MIT

2007 V&V MURI Kickoff
7 August 2006
Problem Scope

Overall Goal:
Develop methods and tools for designing control policies, specifying the properties of the resulting distributed embedded system and the physical environment, and proving that the specifications are met.

Specification
- How does the user specify---in a single formalism---continuous and discrete control policies, communications protocols and environment models (including faults)?

Design and reasoning
- How can engineers reason that their designs satisfy the specifications?
- In particular, can engineers reason about the performance of computations and communication, and incorporate real-time constraints, dynamics, and uncertainty into that reasoning?

Implementation
- What are the best ways of mapping detailed designs to hardware artifacts, running on specific operating systems? What languages are suitable for specifying systems so that the specifications can be verified more easily?
Program Thrusts

Specification and Reasoning Using Graph Grammars
- Build on Klavins’ Computation and Control Language (CCL) & SPIN (Holzmann)
- Use graph grammars to define interaction rules and reason about them

Sum of Squares Techniques (SOS)
- Unified framework for finding invariants and proof certificates for nonlinear and hybrid systems

Extensions
- Probabilistic techniques (specification + algorithms)
- Adversarial settings (including security issues)
- Computational techniques (with JPL/CACR)

Testbeds
- Caltech Multi-Vehicle Wireless Testbed (hardware + sims)
- Alice: 2005 and 2007 DARPA Grand Challenge entry

- Allow temporal logic statements and verification of semi-algebraic conditions to coexist
- Develop design specification and design language plus reasoning tools
Transition Goals and Approach

Goals
- Develop fundamental concepts and tools that get integrated into engineering tools
  - Focus the MURI on specification, design and reasoning (6.1)
  - Utilize partnerships with industry and govt labs to transition to implementation
  - Examples: mutools and SOSTOOLS
- Incorporate feedback from government and industry researchers and practitioners
  - Provide mechanisms for rapid testing of concepts with partners
  - Helps shape research directions to be consistent with most pressing needs

Approach
- Implementation on MVWT and Alice testbeds
- DESTOOLS - Distributed embedded systems toolbox
- Annual workshops and short courses
- Personnel exchange
- Integration with 6.2 and 6.3 efforts
Testbeds

Caltech Multi-Vehicle Wireless Testbed (MVWT)
- Verify cooperative control missions for multiple vehicles
- Build “automated specification and proof interface” for designing missions
- Sample missions: cooperative surveillance, area denial, dynamic reconfiguration

Alice: An Information Rich System for Autonomous Navigation
- Data rich, networked control system: ~1 Gb/s raw data rates, 10 CPU processing
- Representative of level of complexity of UAV and other autonomous systems
Example: RoboFlag Drill (area denial)

**Specification 1** Red Robot Dynamics: \( \Pi_{\text{red}}(i) \)

**Initial:**
\[ x_i \in [\text{min}, \text{max}] \land y_i > \text{max} \]

**Clauses:**
\[ y_i - \delta > 0 : y_i' = y_i - \delta \]

**Specification 2** Blue Robot Control: \( \Pi_{\text{blue}}(i) \)

**Initial:**
\[ z_i \in [\text{min}, \text{max}] \land z_i < z_{i+1} \]

**Clauses:**
\[ z_i < x_{\alpha(i)} \land z_i < z_{i+1} - 2\delta : z_i' = z_i + \delta \]
\[ z_i > x_{\alpha(i)} \land z_i > z_{i-1} + 2\delta : z_i' = z_i - \delta \]

**Specification 3** Assignment Protocol: \( \Pi_{\text{proto}}(n) \)

**Initial:**
\( \alpha \) is a bijection from \( \{1, ..., n\} \) to \( \{1, ...n\} \).

**Clauses:**
\[ \text{switch}_{1,2} : (\alpha(1)', \alpha(2)') = (\alpha(2), \alpha(1)) \]
\[ \ldots \]
\[ \text{switch}_{n-1,n} : (\alpha(n-1)', \alpha(n)') = (\alpha(n), \alpha(n-1)) \]

**Sample drill**
- \( N \) on \( N \) attack, w/ replenishment
- Random initial assignments
- Switching protocol to avoid collisions

**Things that we can prove**
- Semi-automated proofs (Isabelle)
- Avoidance: no two robots collide
- Self-stabilization: if attackers are far enough away, defenders self-stabilize before attackers arrive
Example: Contingency Management for Alice

**Nominal**

No system faults detected

**Slow Advance**

Obstacle ahead on current planned route, approach slowly

**Unseen Obstacle**

Alice prevented from moving forwards by unseen obstacle, mark the area in front as an obstacle in the map

**Lone Ranger**

Verify whether obstacle on current plan really exists

**L-turn Reverse**

Reverse for up to 15m, sufficient to allow Alice to comfortably execute a 90° turn (left or right)

**PLN → NFP**

(obstacle-free terrain)

PLN → NFP through any obstacle type AND Gear == Drive

PLN → NFP through SC obstacle

Alice's attempts to drive through the terrain obstacle have failed → it really exists → mark it as an SC obstacle

PLN → NFP through terrain obstacle only

Alice stationary just in front of terrain obstacle

Alice's speed is < min. maintainable speed but her speed reference (target value) is > the min. maintainable speed AND PLN → NFP (obstacle-free terrain) OR PLN → NFP through terrain obstacle

New SC obstacle created in the cost map in front of Alice's current position now need to back-up to drive around it

Connection not used in race build - if present it allows Alice to escape from dead-end scenarios where she would need to reverse for a distance > her planning distance (30m) by marking dead-ends

After reversing for max L-turn distance: PLN → NFP through SC obstacle, return to Nominal and reassess

**Nominal**

No system faults detected

**Slow Advance**

Obstacle ahead on current planned route, approach slowly

**Lone Ranger**

Verify whether obstacle on current plan really exists

**PLN → NFP**

(obstacle-free terrain)
DESTOOLS: Distributed Embedded Systems Toolbox

Version 1.0 (Year 2)
- Initial implementation of specification & design language (CCL++)
- Automated proof techniques for deterministic specs (SPIN++)

Version 2.0 (Year 3)
- Probabilistic descriptions of specifications, combined with underlying tools

Version 3.0 (Year 4)
- Adversarial descriptions of the operational environment
- Applications of techniques from random algorithms
- Evaluation of scalability to industrial-scale problems

Version 4.0 (Year 5)
- Inclusion of security considerations through adversarial modeling
- Large scale computing support
Annual Workshops and Short Courses

Annual workshops/short courses

- Model after mutools workshops developed by Balas, Doyle and Packard
  - Day 1: tutorial on theory and tools (µ)
  - Day 2: work thru sample application (HIMAT)
- Provide opportunity for researchers to learn about the toolboxes developed under the MURI and apply the design tools to simple problems
- Provide forum for feedback to MURI team and discussion of needed tools
  - Feedback session at end of workshop
- Remote access via web-based forums

Workshop timeline

- Year 1: Caltech - initial spec/design language
- Year 2: MIT - automated proof techniques
- Year 3: ACC - DESTOOLS 2.0 _beta
- Year 4: U. Washington (co-hosted by Boeing?)
- Year 5: ACC - DESTOOLS 4.0 _beta

V&V MURI Kickoff, 7 Aug 06 Richard M. Murray, Caltech CDS
Personnel Exchange

Students and postdocs → Government labs, industry
- Short term visits (3-6 months): student visits to IF, MN, VA, VS + JPL
  - Exploit existing programs at AFRL for hosting visitors
  - Some challenges for grad students with families and international students
- Hiring PhD students
  - Most successful approach for transitioning results and providing feedback

Government and industry visitors → Caltech, MIT, U. Washington
- Short terms visits (1-4 weeks): in-depth discussions and worked applications
  - Example: Sonja Glavaski (Honeywell) visiting Caltech for 1 month; looking at application of SOS to MEMS example
  - Opportunities with Boeing, Honeywell, AFRL (??)
- Longer term, part-time interactions
  - Industry lecturers (eg, Eugene Levretsky, Boeing)
  - Visiting associates (eg, Jeff Rogers, HRL)
  - JPL interactions: lecturer, co-advising, joint research (eg, Gerard Holzmann)
Integration with 6.2 and 6.3 efforts

Large government interest in V&V for software centric systems
- Increasing cost and development time for software-enabled systems
- High level of visibility across DoD and other agencies (eg, NASA/JPL)
- Excellent transition opportunities over next five years across AFRL

Need to link to appropriate 6.2 activities supported by AFRL
- Insure that proper support is available for transitioning results of MURIs to industry
- Example: DARPA SEC program (6.1 results → SEC → flight test)

SBIR/STTR opportunities
- Look for opportunities to create SBIR/STTR solicitations for promising approaches

Synergies between the two MURIs and other 6.1 projects
- Proposals were written without knowledge of each other; synergies are possible
- Opportunities for applying each others’ toolsets (CCL, GME, Ptolomy, SOS, SPIN)
Goals and Agenda

Goals

• Provide an overview of the planned activities and approach of the MURI ✓
• Provide a brief introduction to some of the technologies we will build on ✓
• Describe the strategy for transitioning results to industry and government partners ✓

Agenda

1:45 MURI overview (Murray)
2:15 Specification and Reasoning Using Graph Grammars (Klavins)
2:35 Integrating Continuous & Discrete Domains for Distributed Systems (Chandy)
2:55 Break
3:15 Proof Systems in Continuous and Discrete Domains (Parrilo)
3:35 Complexity Implies Fragility (Doyle)
3:55 Transition Strategy (Murray)
4:15 Discussion and Feedback (Wells/Heise)