Verifying DART Systems (DART)

Presentation to CERDEC
Sagar Chaki
January 15, 2015
1. REPORT DATE  
16 JAN 2015

2. REPORT TYPE  
N/A

3. DATES COVERED  

4. TITLE AND SUBTITLE  
Verifying DART Systems (DART)

5a. CONTRACT NUMBER  

5b. GRANT NUMBER  

5c. PROGRAM ELEMENT NUMBER  

5d. PROJECT NUMBER  

5e. TASK NUMBER  

5f. WORK UNIT NUMBER  

6. AUTHOR(S)  
Chaki /Sagar

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)  
Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213

8. PERFORMING ORGANIZATION REPORT NUMBER  

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)  

10. SPONSOR/MONITOR’S ACRONYM(S)  

11. SPONSOR/MONITOR’S REPORT NUMBER(S)  

12. DISTRIBUTION/AVAILABILITY STATEMENT  
Approved for public release, distribution unlimited.

13. SUPPLEMENTARY NOTES  
The original document contains color images.

14. ABSTRACT  

15. SUBJECT TERMS  

16. SECURITY CLASSIFICATION OF:  
   a. REPORT  
   unclassified  
   b. ABSTRACT  
   unclassified  
   c. THIS PAGE  
   unclassified  

17. LIMITATION OF ABSTRACT  
SAR

18. NUMBER OF PAGES  
14

19a. NAME OF RESPONSIBLE PERSON  

Standard Form 298 (Rev. 8-98)  
Prescribed by ANSI Std Z39-18
Driving Vision

DARTs coordinate physical agents in an uncertain and changing physical world.

- Coordination – physical agents
- Timeliness – safety critical
- Resource constrained - UAVs
- Sensor rich – sensing physical world
- Intimate cyber physical interactions
- Automated adaptation to physical context and rational adversaries
- Computationally complex decisions

Coordination, adaptation, and uncertainty pose key challenges for assuring safety and mission critical behavior of distributed cyber-physical systems.

The DART project uses develops and packages sound techniques and tools for engineering high-assurance distributed CPS.
Currently validated via testing
  • Low coverage, late in development

Rigorous & exhaustive analysis provides higher assurance
  • Non-compositional V&V does not scale
  • Probabilistic & deterministic requirements

Goal: Develop new theories, analyses and tools to engineer high-assurance DARTs with evidence of correctness
DART in a Nutshell

1. Enables compositional and requirement specific verification
2. Use proactive self-adaptation and mixed criticality to cope with uncertainty and changing context

System + Requirements (AADL + DSL) → Verification → Code Generation

Verification:
1. ZSRM Schedulability (Timing)
2. Software Model Checking (Functional)
3. Statistical Model Checking (Probabilistic)

Code Generation:
1. Middleware for communication
2. Scheduler for timing contracts
3. Monitor for functional contracts

Demonstrate on DoD-relevant model problem (DART prototype)
• Engaged stakeholders
• Technical and operational validity
DART High-Level Architecture

Software for guaranteed requirements, e.g., collision avoidance protocol must ensure absence of collisions

Software for probabilistic requirements, e.g., adaptive path-planner to maximize area coverage within deadline

High-Critical Threads (HCTs) | Low-Critical Threads (LCTs)
---|---
MADARA Middleware
ZSRM Mixed-Criticality Scheduler
OS/Hardware

Environment – network, sensors, atmosphere, ground etc.

Node_1

Node_k

Research Thrusts
- Proactive Self-Adaptation
- Statistical Model Checking
- Real-Time Schedulability
- Functional Verification

Validation Thrusts
- Model Problem
- Workbench
# Roadmap & Foundations

<table>
<thead>
<tr>
<th>Thrust Area</th>
<th>Jan</th>
<th>Apr</th>
<th>Jul</th>
<th>Oct</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proactive Self-Adaptation</td>
<td>Latency-aware Self-Adaptation</td>
<td>CMU/SCS FY14</td>
<td>Disaggregation, Machine-learning</td>
<td></td>
</tr>
<tr>
<td>Verification</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Real-Time Schedulability</td>
<td>ZSRM scheduler integrated with DART workbench</td>
<td>HCCPS FY12-FY14</td>
<td>Mixed-criticality among multi-agents &amp; end-to-end OR with Input/Output</td>
<td></td>
</tr>
<tr>
<td>Functional Verification</td>
<td>Bounded Model Checking of Synchronous Software</td>
<td>HCCPS FY12-FY14</td>
<td>Unbounded Model Checking of Asynchronous Software</td>
<td></td>
</tr>
<tr>
<td>Statistical Model Checking</td>
<td>Crude Monte-Carlo based SMC, applied to simple examples</td>
<td>AFOSR FY14</td>
<td>Heterogeneous Fault Regions and Systems with Non-determinism, HPC Simulation</td>
<td></td>
</tr>
<tr>
<td>Workbench</td>
<td>Preliminary version of DSL, Code generation, ZSRM, CBMC, V-REP simulation, simple examples</td>
<td>MCDA FY14</td>
<td>Completed DSL, model problem, ODroid Code Generation, AADL/OSATE, Verification Tools</td>
<td></td>
</tr>
<tr>
<td>Coordination (ELASTIC)</td>
<td>Synchronous, multi-agent</td>
<td>GAMS FY14</td>
<td>Asynchronous, multi-agent</td>
<td></td>
</tr>
</tbody>
</table>
Simple Model Problem: Coordinated Protection

Guaranteed Properties
- No collision

Best Effort
- Defensive perimeter
- Resource conservation (e.g., fewest moves)

Adaptation w/ Uncertainty (next step)
- Lose of a Protector
- Lose of a Leader (new election)
- Directional threats (shield formation vs. perimeter formation)

Assumptions
- 2D Universe (X by Y matrix)
- Perfect communications between agents
- Perfect localization for each agent
- 11 nodes
  - $N_0$ is the leader
  - $N_1$ – $N_{10}$ are the protectors

Operation
- $N_0$ moves from $(x, y) \rightarrow (x', y')$
- $N_1$ – $N_{10}$ move to maintain defensive perimeter
Fleet Operation: Defensive Posture

Free guard UAVs move around to front, simultaneously

Rear guard closes gap, leaving two free guard UAVs

\( N_0 \) moves from \((x, y) \rightarrow (x', y')\)

Coordination needed at each step to avoid collision
Fleet Operation: Defensive Posture

Front guard UAV makes space for $N_0$ to move forward

Free guard UAVs move around to front, simultaneously

Coordination needed at each step to avoid collision
Fleet Operation: Defensive Posture

\[ N_0 \] signals change in direction

\[ N_1 - N_{10} \] comply and begin coordinate perimeter repair

Coordination needed at each step to avoid collision
Mission assurance
- Goals
- Objectives

Resiliency
- Design time Verification
  - Guaranteed behavior
  - Best-effort behavior
- Runtime Assurance
  - Critical Timing behavior
  - Coordination
  - Adaptation
QUESTIONS?
Contact Information Slide Format

Sagar Chaki
Senior MTS
SSD/CSC
Telephone:  +1 412-268-1436
Email:  chaki@sei.cmu.edu

Web
www.sei.cmu.edu
www.sei.cmu.edu/contact.cfm

U.S. Mail
Software Engineering Institute
Customer Relations
4500 Fifth Avenue
Pittsburgh, PA 15213-2612
USA

Customer Relations
Email: info@sei.cmu.edu
Telephone:  +1 412-268-5800
SEI Phone:  +1 412-268-5800
SEI Fax:  +1 412-268-6257