Maturing Assurance Contracts in Model-Based Engineering

NOVEMBER 14–16, 2022

Dionisio de Niz
Technical Director, Assuring Cyber-Physical Systems
Late Discovery of Design Errors in DoD Systems is very costly. Architecture modeling and analysis can detect design error early.

BUT:
Analysis assumptions are often implicit if analysis assumptions not met: analyses break down for reasons not clear to users of analysis tools. E.g., e2e Latency Assumption: periods multiple of each other (harmonic)

DoD barrier for adoption
Digital Engineering: Multiple Claims – Multiple Analyses

Different Assurance Claims

• Combine multiple analysis
• Validate assumptions
• Resolve assumption conflicts

Integrate into arguments to satisfy claims
Analysis Contract: Tracking Assumptions and Guarantees

contract {
  input:
    E2ELatencies

  assumptions:
    areConnectionsDelayed()
    areDeadlinesConstrained()
    areTasksSchedulable()
    areAllThreadsPeriodic()

  analysis:
    meetEndToEndLatencies()

  guarantee:
    [E2EResponses[i] <= E2ELatencies[i] for i in range(len(Responses))]
}

\[ C_L = (A_L, G_L) \text{ with } A_L = \{\alpha_{L,1}, \alpha_{L,2}, \alpha_{L,3}\} \]
Shift Left And Down to the Implementation

Early Analysis
• Evaluate design decisions with partial information
• E.g., latency analysis before periods
  - periods of tasks must be multiples of each other

Refinement
• Track pending information
  - periods
• Track and execute pending verification
  - Schedulability

Conformance
• Track implementation assumption
• Verify implementation conformance
  - Task executed strictly periodic $\alpha_{L,3}$
Assurance Contract Argumentation (1)
Assurance Contract Argumentation (2)
Contract Argumentation in the Development Lifecycle

Integrity of Analysis
• Verify assumptions
  - Detect violation
  - Suggest repairs
• Offer alternative analysis that satisfy assumptions

Refine Design
• When enough new data for new analysis
• When new data affects proof obligations

Argument reusability
• Self-contained modular analysis contracts
Integrity of Analysis: Repairing Assumptions (1)

Contract Verification Plan

Meet deadlines

Claim 2

Claim 3

T2.Period%T1.Period!=0

AADL:
T1.Period=10
T2.Period=15

Harmonic
Sched Bound

Harmonic
Periods

Period=Deadline

RM Priorities

Analysis
Integrity of Analysis: Repairing Assumptions (2)

Contract Verification Plan

Meet deadlines

Claim 2

Claim 3

T2.Period%T1.Period==0

AADL:
T1.Period=10
T2.Period=20

Harmonic Sched Bound

Analysis

RM Priorities

Harmonic Periods

Period=Deadline

guarantee
Integrity of Analysis: Repairing Assumptions (3)

**Suggestion:**
NonHarmonicBound does not need harmonic periods
Refinement Throughout Development (1)

- If no $C, T$, but $u_i = .4$ then find $C_i, T_i$ such that $0.4 = \frac{C_i}{T_i}$, e.g., $0.4 = \frac{4}{10}$

- Utilization: $u_i = \frac{C_i}{T_i}$
- Harmonic Sched Bound
- Analysis
- utilizations $\{u_i\}$
- Enough to execute analysis
- Contract Verification Plan
- Meet deadlines
- Enough to execute analysis
Refinement Throughout Development (2)

- Meet deadlines
- Enough to execute analysis
- Utilization: \( u_i = \frac{C_i}{T_i} \)
- Fixed Priority
- RM Priorities
- Harmonic Periods
- Period=Deadline
- No Priority Inv
- Harmonic Sched Bound
- Analysis
- \( T_{i+1} \% T_i = 0 \)
- \( \{u_i\} \{T_i\} \)
- Proof Obligations
Refrinement Throughout Development (3)

Meet deadlines

Contract Verification Plan

Utilization: $u_i$

$u_i = \frac{C_i}{T_i}$

Fixed Priority

RM Priorities

Harmonic Periods

Period=Deadline

No Priority Inv

NonHarmonic Sched Bound

Response<=$Deadline$

Analysis

utilizations \( \{u_i, T_i\} \)

Proof Oblg

$T_{i+1} \% T_i = 0$

Enough to execute analysis

Proof Oblg
Refinement Throughout Development (4)

- **Meet deadlines**
- **Contract Verification Plan**

- **Utilization**: \( u_i \)
  - \( u_i = \frac{c_i}{T_i} \)

- **Fixed Priority**
- **RM Priorities**
- **Harmonic Periods**
- **Period=Deadline**
- **No Priority Inv**

- **Response <= Deadline**

- **NonHarmonic Sched Bound**

- **Analysis**
  - Utilizations: \( \{u_i\}, \{T_i\}, \{D_i\} \)

- **Proof Oblg**
  - \( T_{i+1} \cdot T_i = 0 \)

- **\( T_i = D_i \)**

- **Enough to execute analysis**
Refinement Throughout Development (5)

Meet deadlines

Contract Verification Plan

Enough to execute analysis

Utilization: $u_i = \frac{C_i}{T_i}$

Fixed Priority

RM Priorities

Harmonic Periods

Period=Deadline

No Priority Inv

Response Time

Analysis

Utilizations: $\{u_i\}, \{T_i\}, \{D_i\}, \{C_i\}$

$T_i = D_i$

$T_{i+1} \% T_i = 0$

$u_i = \frac{C_i}{T_i}$
Refinement Throughout Development (6)

Meet deadlines

Contract Verification Plan

Utilization: $u_i = \frac{C_i}{T_i}$
Fixed Priority
RM Priorities
Harmonic Periods
Period=Deadline
No Priority Inv

Response Time

Analysis

utilizations \{u_i\}, \{T_i\}, \{D_i\}, \{C_i\}

$T_i = D_i$

$u_i = \frac{C_i}{T_i}$

$T_{i+1} \% T_i = 0$

Enough to execute analysis
Argument Modularity

Decomposed into subclaims

Reliable Autobrake
  - Claim 2
  - Claim 3

Defensive Lamar
  - cybersecurity
  - Fault Resilience
  - Meet Radar-Brake Deadline
  - E2E Latency
  - Periodic
  - Delayed Connections
  - Threads Schedulable

Alternatives
  - Harmonic Sched Bound
    - Analysis
  - NonHarmonic Sched Bound
    - Analysis

Incrementally

Utilization: \( u_i \), \( u_i = \frac{C}{T} \)

Fixed Priority

RM Priorities

Harmonic Periods

Period=Deadline

No Priority Inv

Response≤Deadline

\( \sum_{i=1}^{n} u_i \leq 1 \)

Fixed Priority

RM Priorities

Harmonic Periods

Period=Deadline

No Priority Inv

Response≤Deadline

Analysis

Utilization: \( u_i \), \( u_i = \frac{C}{T} \)

Fixed Priority

RM Priorities

Harmonic Periods

Period=Deadline

No Priority Inv

Response≤Deadline

Analysis

\( \sum_{i=1}^{n} u_i \leq 1 \)
Implementation: Symbolic Contract Argumentation

Assumptions
• Constraints that must be satisfied for a valid analysis

Analysis
• Evaluate whether the guarantee can be discharged

Guarantee
• Assertion presented as a true fact on model

Implementation
• Constraint Satisfaction Solver (Satisfiability Modulo Theories – Z3)
• Implements contract argumentation
  - Evaluate whether constrains can be satisfied with facts from analysis guarantees
• Validate assumptions
  - Proof obligations: lack of constraints allow any value that satisfy assumption (e.g., RM priorities)

Artifacts
• Annex language hosted in AADL/OSATE
  - Model Query Language adaptable for multiple modeling language (e.g., SysML V2)
• Automatic Execution of Contract Verification Plan
  - Assumption repair & analyses alternatives
Contract Argumentation Scalability

Exploit Knowledge from Scientific Domain
- Efficient algorithms from specialized domains
  - E.g., greedy worst-case response time in real-time theory
  - Implemented in imperative languages

Assume correctness of analysis
- When validating the contract argumentation
- Enables connection with other lower-level verification results
  - E.g., PROSA: coq (theorem prover) verification of real-time theory

Correctness of implementation
- Exploit proven properties of runtime mechanisms: e.g., schedulers, hypervisors
- Exploit code generation
- Deferred code verification to conform to assumptions
Impact

Certification
• Automated and sound verification of assurance claims through models

Fielding Speed
• Automated assurance watchdog
  - Validate incremental refinement through design
• Concurrent formal assurance argument construction and system development

Digital Engineering and AADL Ecosystem
• Incremental sound analysis infrastructure to DoD modeling efforts
• Architecture-Centric Virtual Integration Practice (ACVIP) within FVL
• DARPA programs using AADL
Concluding Remarks

Certification in Digital Engineering Era
• Follow model-analyze-build
  - Automated argumentation supported by model-based analysis contracts

Shift Left
• Start verifying early design decisions
• As design is refined
  - Ensure properties of previous design decisions are preserved
  - New refinements can provide additional evidence / properties to support assurance

Down To The Implementation
• Drive properties & assumptions down to the implementation

Scalable
• Exploit efficient analysis from different domains

Sound
• Exploit advances in formal verification
  - Combination of analysis, verification of assumptions, implementation compliance, analysis correctness
Maturing Assurance Contracts in Model-Based Engineering
©2022

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

23
Email us at: info@sei.cmu.edu

To learn how the results of our project can help improve the effectiveness of your systems’ assurance.