Late Discovery of System Problems

System integration problems
- System instability and failures
- Implicit and mismatched assumptions
- Shared computing resources
- Complexity of component interaction
  - Functional
  - Extra-functional

Current practice
- Build components first
- Then integrate and test

Way forward
- Analyze system models early and often (Virtual Integration)
- Evolve components and integrated system
MBE offers a way to find more faults in the requirements-architecture design phase

Where faults are introduced
Where faults are found
The estimated nominal cost for fault removal

Model-Based Engineering (MBE) for Computer Based System Architecture

Ensure embedded, real-time system performance and reliability *prior* to system integration, test, or upgrade

Prediction through quantitative analysis & simulation of system operation based on architecture models

System validation through model verification and implementation compliance checking
SAE Architecture Analysis & Design Language (AADL) Standard

Notation for specification of runtime architecture of real-time, embedded, fault-tolerant, secure, safety-critical, software-intensive systems - designed for Model Based Engineering

Fields of application: Avionics, Aerospace, Automotive, Autonomous systems, Medical devices – current focus

Based on 15 years of research & industry input

International Standard approved & published Nov 04, V2 Jan 09

Industry driven standard

www.aadl.info

© 2007 Carnegie Mellon University
Key Elements of SAE AADL Standard

Core AADL language standard (SEI)

- Textual & graphical, precise semantics, extensible

AADL Meta model & XMI/XML standard (SEI)

- Model interchange & tool interoperability

UML profile for AADL – In process (Thales)

- Subset of OMG MARTE profile being defined by MARTE

Error Model Annex as standardized extension (Honeywell)

- Fault/reliability modeling, hazard analysis

Behavior Annex – Draft (Airbus)

- Externally observable behavior of components

Programming Guidelines, Data Modeling Annexes – Draft (ENST)

ARINC 653 Annex – Draft (ENST)
AADL: The Language

Precise execution semantics for components & interactions

- Thread, process, data, subprogram, system,
- Processor, memory, bus, device, abstract component, virtual processor, virtual bus

Continuous signal processing & stochastic event processing

- Data, event, message communication, unqueued & queued
- Synchronous call/return, Shared data access
- End-to-End flow specifications

Operational modes, fault tolerant configurations, levels of service

- Modes & mode transition, error model annex

Modeling of large-scale & configurable systems

- Component variants, packaging of component classifiers, layered systems, parameterized templates, component arrays

Accommodation of diverse analysis needs

- User-defined properties, sublanguage extensions

AADL V2
Single Source Architecture Model

AADL Model

Application

Platform

Timing annotations

Fault annotations

Examples of analyses from same model

Schedulability analysis
Latency analysis

Safety analysis
Reliability analysis

Alternative Hardware Bindings

Low incremental cost for additional analyses & simulations!!!
system GPS
features
  speed_data: in data port metric_speed
    {SEI::BaseType => UInt16;};
  geo_db: requires data access real_time_geoDB;
  s_control_data: out data port state_control;
flows
  speed_control: flow path
    speed_data -> s_control_data;
properties SEI::redundancy => Dual;
end GPS;
System Implementation

system implementation GPS.secure
subcomponents
  decoder: system PGP_decoder.basic;
  encoder: system PGP_encoder.basic;
  receiver: system GPS_receiver.basic;
connections
  c1: data port speed_data -> decoder.in;
  c2: data port decoder.out -> receiver.in;
  c3: data port receiver.out -> encoder.in;
  c4: data port encoder.out -> s_control_data;
flows
  speed_control: flow path speed_data -> c1 -> decoder.fs1
                 -> c2 -> receiver.fs1 -> c3 -> decoder.fs1
                 -> c4 -> s_control_data;
modes none;
properties arch::redundancy_scheme => Primary_Backup;
end GPS;
Some Standard Properties

Dispatch_Protocol => Periodic;
Period => 100 ms;
Compute_Deadline => value (Period);
Compute_Execution_Time => 10 ms .. 20 ms;
Compute_Entrypoint => “speed_control”;
Source_Text => “waypoint.java”;
Source_Code_Size => 12 KB;

Thread

Thread_Swap_Execution_Time => 5 us.. 10 µs;
Clock_Jitter => 5 ps;

Processor

Allowed_Message_Size => 1 KB;
Propagation_Delay => 1µs .. 2µs;
Bus_Properties::Protocols => CSMA;

Bus

Protocols is a user defined property
Readable Graphical Specification
Flight Manager in AADL

From Partitions

Nav signal data

Nav sensor data

20Hz

Navigation Sensor Processing

10Hz

Integrated Navigation

Nav sensor data

Nav data

Guidance Processing

Nav data

20Hz

Guidance

Flight Plan Processing

5Hz

FP data

Nav data

FP data

Aircraft Performance Calculation

2Hz

Performance data

Fuel Flow

To Partitions
Architecture Execution Semantics Defined – Components to SoS

Nominal & recovery
Fault handling
Resource locking
Mode switching
Initialization
Finalization

Thread Example Diagram
PCS Technical Detail: *Single-Model, Multi-Dimensional Analysis*

**SECURITY**
- Increased confidentiality requirement
  - change of encryption policy

**ARCHITECTURAL MODEL**
- Key exchange frequency changes
  - Message size increases
    - increases bandwidth utilization
    - increases power consumption

**RESOURCE CONSUMPTION**
- Bandwidth
- CPU Time
- Power Consumption

**REAL-TIME PERFORMANCE**
- Deadlock/Starvation
- Latency
- Execution Time/Deadline

**Increased computational complexity**
- increases WCET
- increases CPU utilization
- increases power consumption
- may increase latency

Confidence
Rapid Growth, Diversity of AADL Toolsets

OSATE – Open Source – Editor with analysis

- SEI developed, full language editing and semantic checking, multiple analysis plug-ins, Eclipse based, integrated text and graphical editing with TOPCASED

TOPCASED – Open Source – Model Bus Framework for integration of tools and methods

- Airbus led, 20 companies, Metamodeling Framework, AADL Graphics, AADL XML, model transformation, Behavior Annex, also will support UML, stable July 2007, includes new tools from SPICES.

STOOD – Commercial – Development support, Editor, Analysis

- CASE toolset supporting UML, HOOD and AADL. Includes transformations between notations, document support, requirements support. Works with OSATE, TOPCASED, OCARINA. Includes AADL simulator, Cheddar scheduling analysis.

OCARINA – Open Source – Middleware generation and system integration

- ENST AADL graphics and middleware generation and integration to AADL model of network distributed processors. Creates formal model of executive integrated in AADL. Generates to network protocols – CORBA, RT, FT

Fremont – Open Source, Formal analysis based tools, consulting and OSATE support

- AADL to ACRS (process algebra), formal analysis of concurrent resources, AADL to Charon, generation and integration of hybrid control systems, AADL Architecture Simulator – integrates event driven and schedule driven

CHEDDAR – Open Source – Scheduling analysis

- Univ of Brest, advanced scheduling analysis toolset

EDICT – Commercial – Fault Tolerant Systems and Security Analysis

- WWTechnology – Error handling, Safety and Information Assurance modeling using AADL

Consortium and Company Owned – SPICES, AVSI, ASSERT plus internal integrations
AADL IMA System Analysis for Trade Studies
Display System Graphical AADL

Original Configuration from AADL

New Configuration

Notes:
Identifiers with angle-bracketed terms are replicated for each unique set of terms, where terms are defined as:
<cpm>: Common processing machine name
<cpmid>: Longer name of cpm
<ndo>: Network data object name
<sw>: ASL switch side identifier
<vmr>: Virtual machine and rate, indicating thread name

Automated Analysis
5 CPMs
13 Virtual Machines
90 Threads
165 End-End Flows

Source: John Mettenburgh
Rockwell Collins
Evaluations of various methods and tools have been carried out over the past few years using one or more of the following workloads.

**Air transport aircraft IMA (simplified production workload)**
- Globally time-triggered
- 6 processors, 1 multi-drop bus
- 105 threads, 51 message sources

**Military helicopter MMS (first release, partial)**
- Globally time-triggered
- 14 dual processors, 14 bus bridges, 2 multi-drop buses
- 306 threads, 979 [source, destination] connections

**Air transport aircraft IMA (preliminary, partial)**
- Globally asynchronous processors, precedence-constrained switched network
- 26 processors, 12 switches
- 1402 threads, 2644 [source, destination] connections

**Regional aircraft IMA (production workload)**
- Globally time-triggered
- 49 processors, 2 multi-drop busses
- 244 processes (TBD threads), 3179 [source, destination] connections

Source: Steve Vestal
Honeywell
Industrial Embedded Systems Initiatives

- **Automotive**
  - EAST ADL AutoSAR
  - US AVSI Avionics Consortium
    - Analysis-based System Validation
    - 8 partners $12+M 2007-2010
  - ITEA SPICES
    - Model-Driven Embedded Systems Engineering
    - 15 partners €16M 2006-2009
  - TOPCASED
    - Open Source Embedded Systems Tool Framework
    - 28 partners €20+M 2005-2008

- **Avionics**
  - MBE
    - AADL
      - SAE AADL Standard
        - Nov 2004
      - AADL Meta Model & XMI
        - June 2006
      - AADL Error Annex Standard
        - June 2006
      - OSATE Toolset
        - SEI
  - EC ASSERT
    - Proof-based Satellite Architectures
      - ESA + 30 partners
      - €15M 2004-2008

- **Aerospace**
  - OpenGroup Real-Time Forum
    - EU + US partners
  - IST ARTIST2
    - Embedded Systems Center of Excellence
      - 2007-2011
Embedded Systems (EMS) Life Cycle Activities and Problems

**Assert Focus**

- **System Engineering**
- **Software/Hardware Development**
- **Integration**

**Existing Methods**
- Formal methods

**Existing Practice**
- Intensive activity on real elements to test benches

**Family Engineering**
- **Development of Building blocks**
- **Validation of the Family**

**Proof Based System Engineering**
- Architecture Description Language
- Formal SW Eng.

**Methodological Features**
- Mastering the complexity once
- Proving the basics of the family once
- Full coverage of EMS development cycle
- Multi-domains and Dissemination are essential

**Benefits**
- Production of a new system from family
- Schedule and cost efficient
- Full confidence in proofs at family level
- Proofs are kept for each instance

**NO System engineering method**
- Empirical approach

Validation obtained since engineering phase

Tailoring keeps the proof validity
SPICES Work Packages (3 of 5)
17 EU Companies, 3 yrs, 16M Euro, Starts Sept 2006

Tools Process Pilots

EXTENDED AADL

AADL MODELLING TOOL

WP1

SYSTEMC GENERATOR

VALIDATION & VERIFICATION TOOLS

WP2

SYSTEMC CODE

WP3

APPLICATION CODES (XML)

CONTAINER GENERATOR

DEPLOYABLE COMPONENTS & CONTAINERS

ADMINISTRATION

EXECUTION

(SystemC)

(CCM)
Demonstrating A Model-Based Acquisition and Development Process

Who: Avionics Industry Consortium (AVSI)

• Airbus, Boeing, LMCO, Rockwell-Collins, BAE Systems, GE Aviation, …plus FAA and DoD

Why System Architecture Virtual Integration (SAVI)

• “We cannot create the next generation airplanes with current methods”

As-is and to-be acquisition & certification processes

ROI model driven by life cycle error data

Proof of Concept

• Integrator-subcontractor interaction through architecture model
• Migration of functionality between hardware and integrated modular avionics (IMA) software
• Scalability to full system
• Single source reference model & model repository
• Early error discovery
- Req – Arch design phase: 60 (of 70) of faults introduced are detected (change from 45.5)
- Integration phase: Reduction in detection from 50.5 to 10 (change from 20)
- Acceptance test: Reduction in detection from 9 faults to 4.5 faults
- Number of faults detected by user is further decreased from 20 to 5 (change from 10)

Reduction of faults in code development from 20 to 2 faults.

46.8% reduction
50.3% reduction
71.8% reduction

Number of faults: 100
Fault removal cost: $x$ for one fault introduced in the same phase
y-axis represents cumulative cost over the phases
Benefits

Model-based system engineering benefits

Improved predictability early and throughout life cycle greatly reduces integration and maintenance effort

Benefits of AADL as industry standard

AADL as standard provides confidence in language stability, leveraged technology & training investment across organizations
Predictive Analysis and Generation

Modeling notations/methods can complement according to their purpose

- SysML for system engineering: physical system characteristics and concepts of operation
- UML for conceptual and detailed SW design: platform independent SW modularity for reuse
- SAE AADL for software system engineering: platform-specific models of run-time architecture

AADL analysis & generation tools

- Based on strong semantics in modeling language
- Single source architecture model supporting multiple perspectives
- Extensible language to address range of lifecycle incremental analysis
- Leverage integrated analysis and generation tools for rapid prototyping
Wins Of SAE AADL

Static & dynamic architecture in single model
  - Improved software process
Validation based on precise semantics
  - Validated system architectures
Common architecture notation
  - Sub-contractor management
Standarized interchange format
  - Tool integration & interoperability
Alignment with UML2.0
  - UML profile, OMG MARTE
1995 MBE Missile Re-Engineering Demo - Cost and Schedule Benefits

AMCOM SED studies show
- NRE cost/schedule reduction for first delivery
- NRE cost/schedule reduction for upgrades

Application of methods and tools that determine resource requirements.
- Efficiency of allocation and scheduling
- Degree of redundancy needed for safety and availability

Biggest savings may be in reduced recurring cost
- Recurring cost exceeds NRE over product lifetime (except space)
- Easier upgrade means faster technology refresh & upgrade
- Reduced size/weight/power ripples through rest of vehicle

total project savings 50%, re-target savings 90%
Model-based Assurance

Predictive Analysis Across Perspectives

Availability & Reliability
- MTBF
- FMEA
- Hazard analysis

Security
- Intrusion
- Integrity
- Confidentiality

Data Quality
- Data precision/accuracy
- Temporal correctness
- Confidence

Real-time Performance
- Execution time/Deadline
- Deadlock/starvation
- Latency

Resource Consumption
- Bandwidth
- CPU time
- Power consumption

Reduced model validation cost due to single source model

Software Engineering Institute | Carnegie Mellon
© 2007 Carnegie Mellon University
Analysis of Operational Characteristics

Useful from many current approaches

- UML
- MARTE
- SysML
- Assurance Cases
- Existing Architecture
- Simulink
- Domain Component generators
- Product Line
- Reference Architecture
- Arch Tradeoff Analysis M
- New Approach Virtual Integration
- AADL Model of Runtime Architecture
- Error Model Annex
- Behavior Annex
- UML Annex
- ARINC653 Annex
- Analysis based on precise semantics
- In Progress

© 2007 Carnegie Mellon University
How does AADL fit with Assurance Case?

An assurance case is

- a documented body of evidence
- that provides a convincing and valid argument
  - showing that an artifact
  - has desired properties
- i.e., a documented chain of reasoning (based on evidence) that supports some claim

Assurance cases are used to explain why one should have confidence that some claim holds

AADL and MBE provides an architectural means of quantitatively making claims (incrementally building the case) and validating the impact of interactions across the architecture as changes are made.
SPICES Technological approach

Tools
- Requirements
- Architecture modelling
- Platform Modeling
- Verification
- Model simulation
- Real-time Schedulability analysis
- Power-consumption analysis

AADL

CCM

SystemC

Execution
- Space Pilot
- Avionics Pilots (5)
- Software radio platform Pilot
- Simulation
- Implementation
- Timing prediction
- Verification

Quality standards
- DO-178B
- DO-254

Real-time Integration support
Power-mngt

POSIX RTOS
Partitioning Kernel (ARINC-653)
GPP, DSP, FPGA
SystemC
C, C++ VHDL (SystemC), Matlab/Simulink
Static code analysis

SEI Presentation (Full Color) Bruce Lewis, 5/28/2008
© 2007 Carnegie Mellon University
AVSI SAVI Modified Business Model

System Integrator defines a new product using internal repository of virtual “parts”

Specifications for virtual subcomponents sent to suppliers

Once virtually integrated -> Build
AVSI SAVI Overview of Multi-Aspect Model Repository & Model Bus

Requirements

- Eclipse
- MatLab
- SimuLink
- Rhapsody
- Esterel
- SysML

Model Repository

AADL

Verification

- OSATE
- SCADE
- TOPCASED
- DOORS

Model Bus

Design

Integration/Deployment

SEI Presentation (Full Color)
Bruce Lewis, 5/28/2008
© 2007 Carnegie Mellon University
MBE SEI Transition Support

IR&D – Experts in advanced architectural concepts and analysis
  - Networked Sensors Modeling – 2006
  - Security plus MILS Modeling – 2007-2008
  - Processor Performance Modeling - 2007
  - Fault-Tolerant Systems, Multi-Core – 2008

Training – Experts in AADL and MBE
  - Concepts, Language and Tools, Start-Your-Architecture Workshop

Consulting – Team support

Open Source Tools with Analysis to Support
AADL Components

Application Software
- thread
- thread group
- process
- data
- subprogram

Execution Platform
- processor
- memory
- bus
- device

Composite
- system

Each component has predefined properties associated with its declaration.