Ground systems modeling using the Architecture Analysis & Design Language (AADL)

Dr. Michela Muñoz Fernández
NASA Jet Propulsion Laboratory, California Institute of Technology

February 25, 2014

MBSE FOR GDS

- Use of Model-Based Systems Engineering (MBSE) to enable more robust and complete systems engineering and integrated analysis of complex System-of-Systems (SoS) problems which have historically been implemented via paper/ presentation-based design capture, disparate models, in documents, and in the brains of expert engineers across many disciplines
- Can new tools and technologies be used in future missions starting at earlier phases to reduce risk?

The Architecture Analysis & Design Language

- The SAE Architecture Analysis & Design Language (AADL) is an architecture description language for real-time, fault-tolerant, scalable, embedded, modular multiprocessor systems.

- AADL enables the development of highly evolvable systems, early and quantitative analyses of a system's architecture, and evolution of an architecture model for continued analysis throughout the lifecycle.

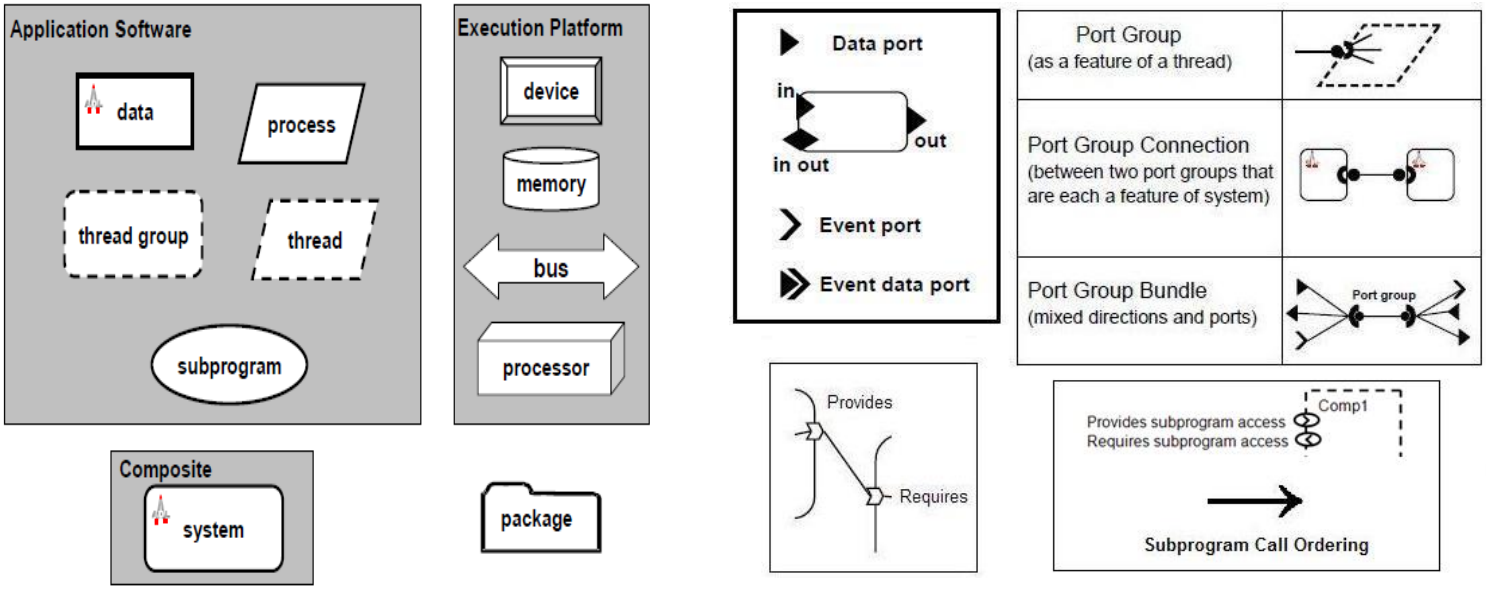
- Customers: System architects that would like to optimize the decision on system architectures and/or any engineer in general that would like to model embedded systems

The Architecture Analysis & Design Language

- AADL's capabilities:
  - Create and analyze component-based models of a task and task interaction architectures of embedded software
  - Predictive analyses of operational characteristics (meeting deadline, response time, and throughput requirements)
  - Discover system integration problems early in a development effort

Using AADL

- **Using AADL and OSATE**
  - OSATE → Open Source AADL Tool Environment

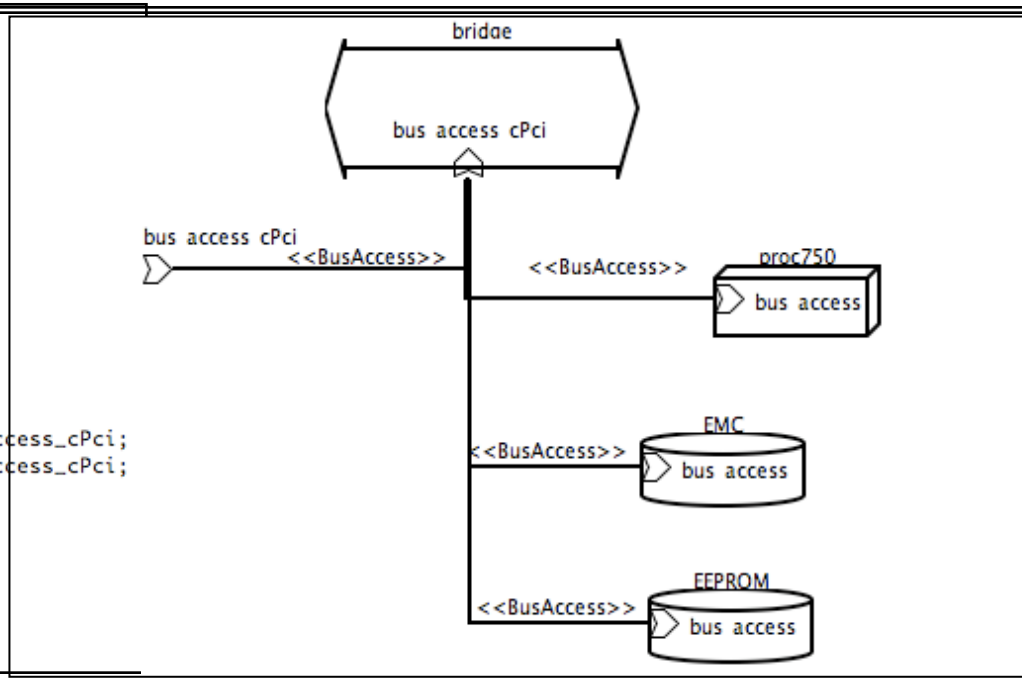


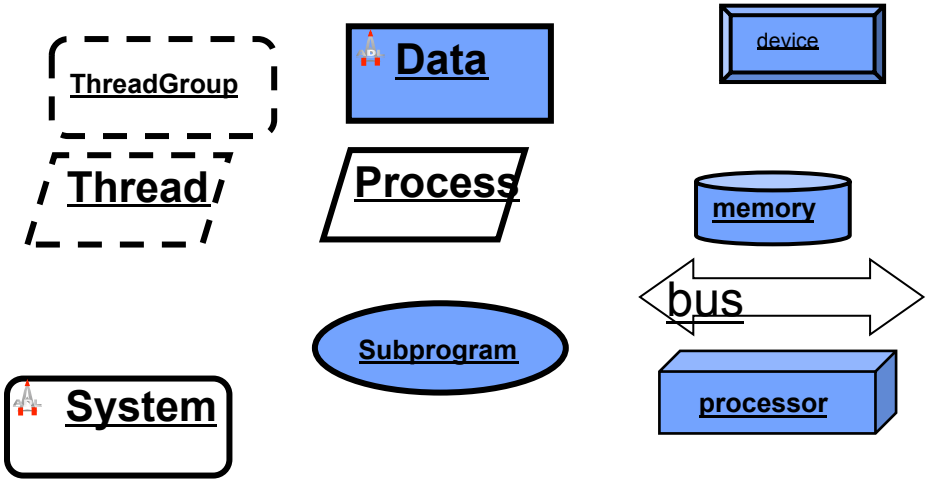
Source: http://www.aadl.info

5

AADL Textual and graphical representation

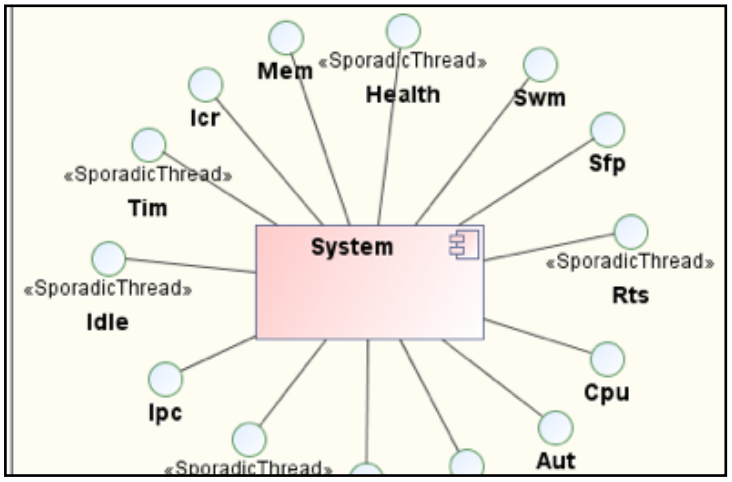
```
system implementation board750.msap
  subcomponents
    bridge: bus bridge;
    proc750: processor proc750;
    EEPROM: memory EEPROM;
    EMC: memory EMC;
  connections
    bus_access_01: bus access bridge -> proc750.bus_access;
    bus_access_02: bus access bridge -> EEPROM.bus_access;
    bus_access_04: bus access bridge -> EMC.bus_access;
    BusAccessConnection1: bus access bus_access_cPci -> bridge.bus_access_cPci;
    BusAccessConnection2: bus access bus_access_cPci -> bridge.bus_access_cPci;
end board750.msap;
```

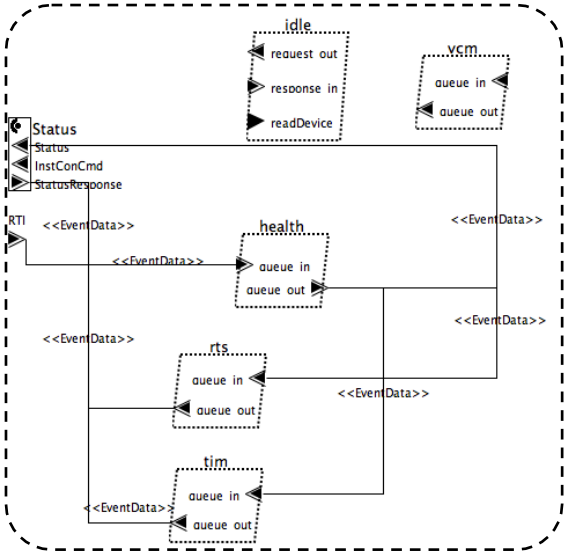




MBSE FOR GDS

- **Ground systems software has been developed without characterizing *performance* of the real-time system being built until integration**
- **Finding execution-related issues at that point is costly**
  - **AADL** (Architecture Analysis and Design Language) **model shows execution interactions between high-level system components**
    - Enables early quality attribute analyses
  - **AADL reduce possibility of doing rework *later* in the lifecycle**
    - Increases confidence at gateway reviews, by providing independent, semantically accurate analyses





AADL SysML Comparison

- AADL was born as an avionics-focused domain-specific language and later on was revised to represent and support a more general category of embedded real-time systems
- SysML is an extension of the Unified Modeling Language (UML) intended to support modeling system engineering applications
- SysML focuses on the “big picture” architectural views, whereas AADL addresses the more detailed platform-oriented and physical aspects of such systems

AADL SysML Comparison

- Mutually complementary:
- SysML: standarized language for systems engineering. Provides support for requirements engineering, traceability, and precise modeling of diverse physical phenomena.
- AADL: oriented towards the modeling of real-time embedded systems and includes a comprehensive catalogue of hardware and software elements common in such systems and their characteristics, allowing relatively precise and dependable analysis of different system properties such as performance, timing, or power consumption

AADL SysML Comparison

SysML

Requirements, Traceability, Parametric models, Interactions

AADL

Quantitative Analysis, Hardware-Software Component Categories, Software to Hardware Binding

SysML and AADL

Modes/State Machines, Components/System Blocks, Component Interactions/Block Flows

AADL for NASA GDS

- The NASA Exploration Technology Development Program develops long-range technologies to enable human exploration beyond Earth orbit and also integrates and tests advanced exploration systems to reduce risks and improve the affordability of future missions

Human exploration beyond Earth orbit.
Image credit: NASA.gov

Autonomous Systems and Avionics:

-Develops and demonstrates integrated autonomous systems capable of managing complex operations in space to reduce crew workload and dependence on support from Earth

-Technologies will address operations in extreme environments, efficient ground-based and on-board avionics systems and operations, and cost-effective human-rated software development

Source: http://go.nasa.gov/groundsystems

AADL capabilities will help:

Real-time software more prevalent in avionics systems (unmanned aerial vehicles, spacecraft):

-Operation with limited human interaction or an increased latency in human response

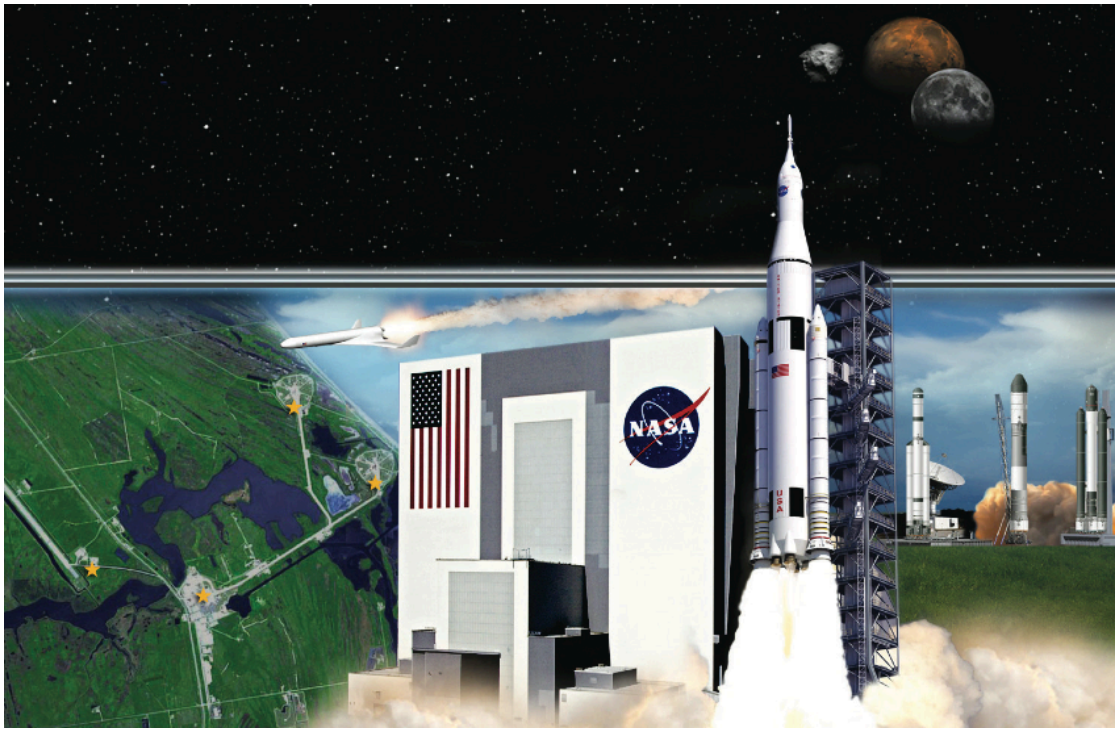
-Autonomous fault detection and repair capabilities are required to respond to off nominal conditions that are encountered

Future and current needs for NASA GDS

- Ground Systems transformation from a historically government-only launch complex to a spaceport bustling with activity involving government and commercial vehicles
- Updated center to process and launch the next-generation vehicles and spacecraft designed to achieve NASA's goals for space exploration:

-Developing the necessary ground systems while refurbishing and upgrading infrastructure

-Government, commercial and other customers



Source: http://go.nasa.gov/groundsystems

Examples of applications for NASA GDS

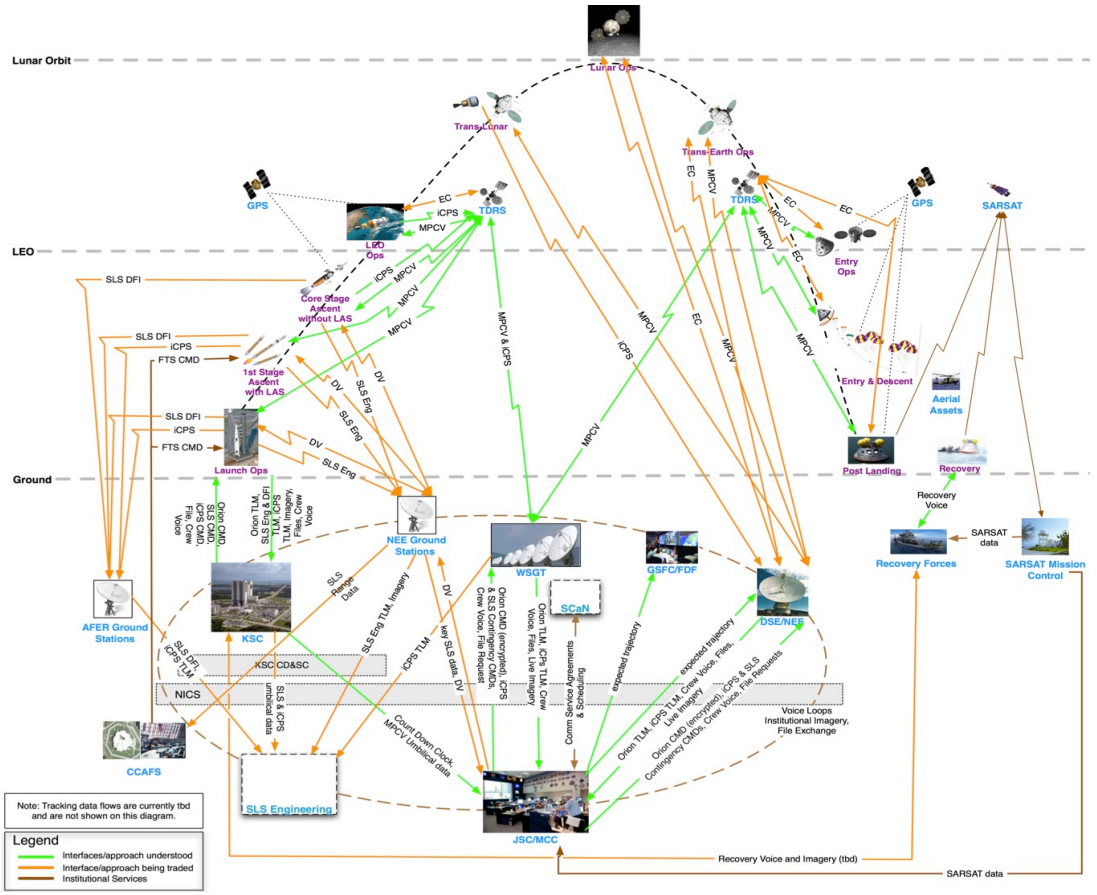
High fidelity End-to-End data delivery model with AADL for the Orion mission

Unmanned test flight -Exploration Flight Test-1 or EFT-1. Image Credit: NASA.gov

-Implementation of an End-to-End data delivery model using the Architecture Analysis and Design Language (AADL) including:
  -S/C data generation process taking into account onboard memory, latency and link capability
  - Downlink process
-End-to-End throughput and latency analysis
-Investigate system to system interdependencies

End-to-End Mission System modeling for the Orion project

- Each mission phase requires a specific configuration of systems, required data exchanges and communications links between the different systems (e.g. on pad operations differ significantly from flight operations)



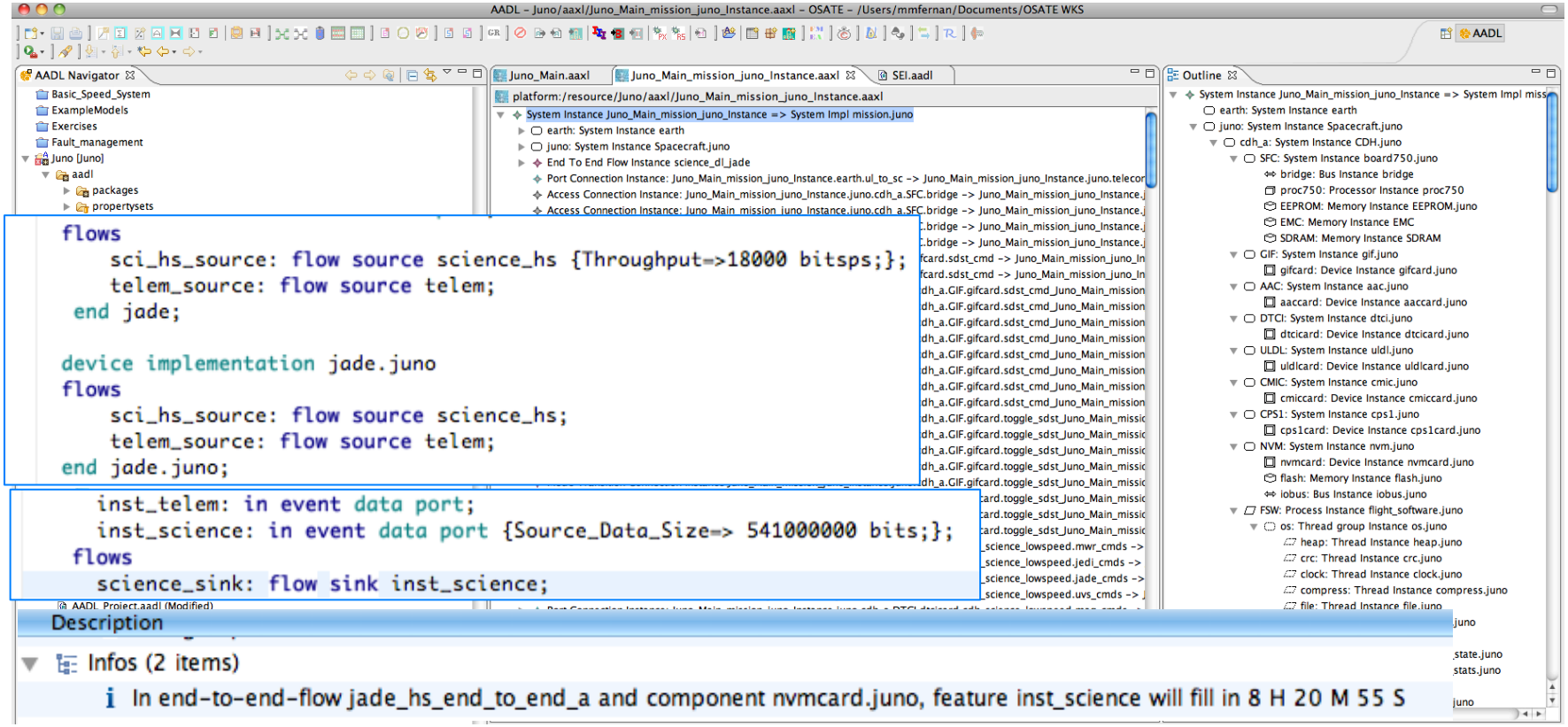
Source: Architecting the Human Space Flight Program with Systems Modeling Language (SysML). Maddalena Jackson, Michela Muñoz Fernández, Oleg Sindiy, Thomas McVittie. *AIAA* Infotech@Aerospace. 19 - 21 June 2012.

End-to-End Mission System modeling for the Orion project

- AADL's capabilities enable modeling of systems comprised of hardware and software subsystems connected to each other via hardline and RF communications links that support the exchange of critical data such as Commands (CMD), various forms of Telemetry (e.g., Operational, Developmental, Engineering), File Exchanges, Primary and Dissimilar Voice, Video/Motion Imagery, Time
- The configuration of the systems, required data exchanges and communications links may change significantly between mission phases

AADL modeling of Juno mission

- Demonstrated the use of the AADL Practice Framework:



By modeling the Juno spacecraft and applying new tools, some command errors could have been revealed in real time

Juno's final integration of the instruments onto the S/C revealed that: the "pipeline" of software that received data from Juno's instruments converted it into product files, and sent it to the Instrument teams was not optimized. Earlier modeling would have detected this situation before integration

AADL Error Annex

-Enables modeling of different types of faults, fault behavior of individual system components, modeling of fault propagation affecting related components in terms of peer to peer interactions and deployment relationships between software components and their execution platform, modeling of aggregation of fault behavior and propagation in terms of the component hierarchy, as well as specification of fault tolerance strategies expected in the actual system architecture

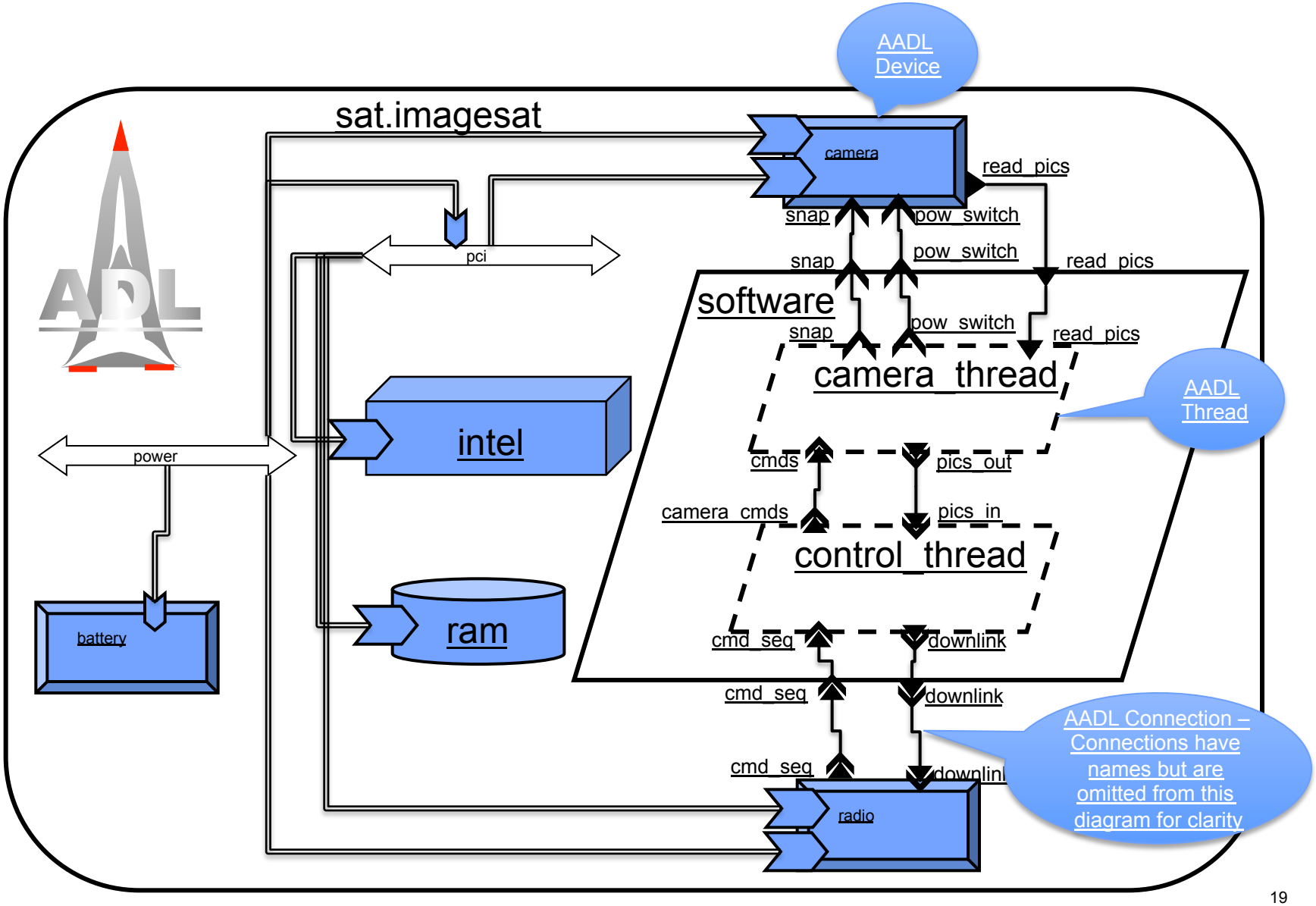
Supports qualitative and quantitative assessments of system dependability, i.e., reliability, availability, integrity (safety, security), and survivability, as well as compliance of the system to the specified fault tolerance strategies from an annotated architecture model of the embedded software, computer platform, and physical system

Error propagation with AADL

JPL

- Errors can propagate between software components and execution platform components they are bound to.
  - The keywords processor, bus, virtual processor, virtual bus, memory, and device are used to identify the binding point of a software component with the execution platform component it is bound to
  - The keyword binding is used for connections and virtual buses to identify their binding to execution platform components.
  - The keyword bindings is used in execution platform components to identify the binding point of components bound to them. Propagations with respect to bindings can be in both directions.

Spacecraft with an imager



Conclusions

- **AADL potential for MBSE applied to NASA GDS**
  - Already some built-in plug-ins that allow multiple analyses
  - Reason about the system and answer questions such as
    - is there enough data bandwidth?
  - CPU resource analysis
  - Do we have enough CPU margin? (does it meet the requirement for a specific mission phase?)
- Performance analyses:
  - Bus Bandwidth Analysis
  - Memory Resource Analysis
  - Deadlock Analysis (UPPAAL)
  - Reachability Analysis (UPPAAL)
- Applications to cloud computing
- Applications to cyber security

Conclusions

- The Architecture Analysis and Design Language (AADL) is a Society of Automotive Engineers (SAE) standard notation (AS5506/1) for the modeling and analysis of real-time systems. AADL has been extended to model fault management behavior through the AADL Error Annex, also an SAE standard

- The Architecture Analysis and Design Language and the AADL Error Annex can be used to assure dependability in the software fault management system in an avionics, real-time embedded systems

- AADL's applications to software assurance. The approach needs to be supported by a standard, repeatable framework. The foundation for this framework is in the ability to model the fault management system integrated in the hardware and software avionics real-time system. Models can assist in assuring both functional and quality attribute requirements such as reliability. NASA Office of Safety and Mission Assurance (OSMA) Software Assurance Research Program (SARP)

References

- "Software Assurance Standard," NASA-STD-8739.8 w/Change 1. 2004.
- Barbacci, Mario, Mark H. Klein, et al. "Quality Attributes." CMU/SEI-95-TN-021. 1995.
- Vestal, Steve, Larry Stickler, Dennis Foo Kune, Pam Binns, and Nitin Lamba. *AADL - Documents*. Honneywell,16 June 2004. Web. <www.aadl.info/aadl/documents/AADL-MetaH%20for%20LAS.pdf>.
- Feiler, Peter H., David P. Gluch, and John J. Hudak. *The Architecture Analysis & Design Language (AADL): An Introduction.* CMU/SEI-2006-TN-011. 2006.
- OSATE: Open Source AADL Tool Environment. http://www.aadl.info. October 2009.
- SAE Embedded Computing Systems Committee. SAE Architecture Analysis and Design Language (AADL) Annex Volume 1: Annex A: Graphical AADL Notation, Annex C: AADL Meta-Model and Interchange Formats, Annex D: Language Compliance and Application Program Interface Annex E: Error Model Annex. As-2c Architecture Analysis And Design Language. 2006.
- NASA Ground Systems Development and Operations Program. http://gonasa.gov/groundsystems

References

- Evensen, Kenneth D., Michela Muñoz Fernández. "Assuring Software Fault Management with the Architecture Analysis and Design Language." *AIAA* Infotech@Aerospace. 19 - 21 June 2012. Garden Grove, CA.
- Evensen, Kenneth D. and Dr. Kathryn Anne Weiss. "A Comparison and Evaluation of Real-Time Software Systems Modeling Languages." *AIAA Infotech@Aerospace 2010.* Atlanta, GA, Apr. 20-22, 2010.
- Feiler, Peter H., and Ana Rugina. "Dependability Modeling with the Architecture Analysis & Design Language (AADL)." CMU/SEI-2007-TN-43. 2007.
- Heimerdinger, Walter L., and Charles B. Weinstock. "A Conceptual Framework for System Fault Tolerance." CMU/SEI-92-TN-033. 1992.
- Architecting the Human Space Flight Program with Systems Modeling Language (SysML). Maddalena Jackson, Michela Muñoz Fernández, Oleg Sindiy, Thomas McVittie. *AIAA* Infotech@Aerospace. 19 - 21 June 2012. Garden Grove, CA.

Questions?