A MODEL-DRIVEN SOFTWARE COMPONENT FRAMEWORK FOR FRACTIONATED SPACECRAFT

Abhishek Dubey(1), Aniruddha Gokhale(1), Gabor Karsai(1), William R. Otte(1), Daniel Balasubramanian(1), Sandor Nyako(1), Johnny Willemsen(2)

1: Institute for Software-Integrated Systems, Vanderbilt University, 1061 16th Avenue South, Nashville, TN 37212, USA, +1(615)343-7472, abhishek.gokhale,gabor,otte,daniel,nyako@isis.vanderbilt.edu
2: Remedy IT, 2830 AC Berkell en Rodenburg, The Netherlands, johnny@remedy.nl

DEFINITION

A fractionated spacecraft is a satellite architecture where the functional capabilities of a conventional monolithic spacecraft are distributed across multiple modules which interact through wireless links. [Wikipedia]

NEEDS

1. Software platform with support for resource sharing, security, and fault tolerance
2. Software toolchain for modeling, synthesis, analysis, and verification

PLATFORM PROPERTIES AND REQUIREMENTS

Distributed computing system with network addressability
- Fully distributed platform with fluctuating connectivity

Dynamic architecture
- Modules of the cluster, software apps running on the platform, labels on information flows, location of software and hardware resources used can (be) change(d) at any time

Secure resource sharing
- Resources must be securely shared across applications: processor, communication links, memory, software services

Fault tolerance
- Faults in apps, services, comm. links, computing and spacecraft hardware are detected, isolated, and their effects mitigated

GOAL:

Tool-supported, effective software development process and system integration.

This work was supported by the DARPA System F6 Program. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of DARPA.

MIDDLEWARE LAYER - INTERACTIONS

1. Synchronous and asynchronous point-to-point interactions with call/response semantics (Subset of OMG CORBA RMI/AM)
   - Location transparency
   - Request (de)multiplexing
   - Message (de)marshaling
   - Error handling
   - Support for QoS (client timeouts, reliable one-ways)

2. Anonymous publish/subscribe interactions with one/many-to-many data distribution patterns (Subset of OMG DDS)
   - Datatype specification
   - Static discovery
   - Selected QoS: reliability, time-based filtering, latency budget, etc.

COMPONENTS

A component is a self-contained, potentially many-objects, The component is parameterized, has state, consumes resources, publishes and subscribes to topics, provides interfaces to and requests interface from other components. A required interface (publisher does not wait)

Required (interface)
- Required (publish)
- Synchronous communication using call-response semantics
- Triggering can be time-driven or event-driven

PARAMETERS

- Parameter
- Required (interface)
- Provided (interface)
- Static
- Dynamic
- Static
- Required (interface)

COMPONENT OPERATIONS

- Single threading: Every component has a single-thread - only one operation is active in a component at any time.
- When an operation becomes eligible (await of eventcall, other, etc.), it is placed into the scheduling queue of the Operating system.
- A thread that handles the operation requires a scheduling policy (priority FIFO or EDF) and runs in the kernel thread.
- Resource allocation, deadline is monitored and enforced

MODEL-DRIVEN DEVELOPMENT

- Code IDE
- Deployment
- V&V

System Models

- Model-based Application
- Component-based Application
- Distributed Deployment
- Interacting Components

Summary

- Fractionated spacecraft = distributed real-time embedded system
- Developers need design-time and run-time tools to create and integrate complex applications
- Component-driven development: Applications are constructed from independently developed and verified, reusable components that participate in well-defined interactions
- Information Architecture Platform = Run-time system that provides a framework for software components
- Model-driven development = Explicit models of the application architecture, hardware platform and deployment infrastructure code, deployment plan, etc. are generated