Integrated Communications Systems Modeling

WInnComm 2014

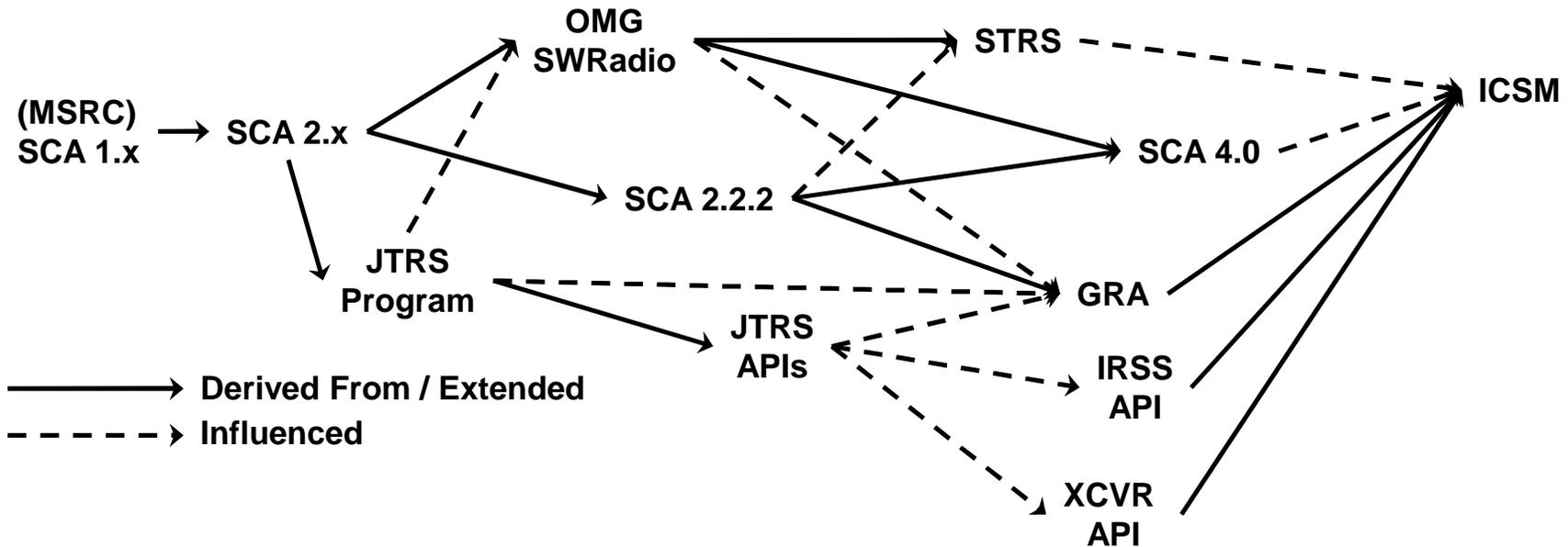Vince Kovarik

PrismTech

SDR forum
version 2.0

Objectives

- **Focus on the integration of hardware and software architecture specification for communications systems.**

- **Leverage prior work in SDR standards and systems to:**

  - Develop an Open Modular Radio Architecture using industry standard modeling languages and tools.

  - Promote the re/use of the architecture through model extensions and specialization.

  - Provide a reference architecture in the form of a SysML/UML model.

- **Promote the adoption and use of the Open Modular Architecture within the forum and industry.**

WIRELESS
INNOVATION
FORUM™   Driving the future of radio communications and systems worldwide

SDR forum version 2.0

Rationale

- **Much of the work to date in SDR type systems has focused on a particular aspect of the system such as the software infrastructure.**

- **The hardware architecture has been traditionally performed as a semi-independent process.**

- **This approach can lead to integration and performance issues.**

- **This WInnF project focuses on developing an extensible integrated systems model.**

WIRELESS INNOVATION FORUM™

Driving the future of radio communications and systems worldwide

SDR forum version 2.0

Evolution of Standards and Specifications

- **SCA was the foundation specification of the JTRS program**
- **Objective of the OMG effort was to establish SCA as an industry standard**
- **STRS targeted for space deployed systems**
- **GRA originally started as alternative to SCA for A2G systems, e.g. SATCOM, with focus on Systems Model**
- **ICSM will leverage GRA work to provide a reference design model for communications systems**

WIRELESS INNOVATION FORUM™
Driving the future of radio communications and systems worldwide

SDR forum version 2.0

Focus Area of Standards

Communications Systems Model / GRA

SCA 4.0 / OMG Base Spec **SCA 4.0 / OMG Appendices**

SCA 2.x / STRS

ICSM Project Focus **ICSM Follow-on Projects**

CIM   PIM   PSM   PSI

System Requirements
Black Box Behaviors
Use Cases
Functional Block Diagram

System Architecture
Activity Diagrams
Functional Decomposition
Interface Descriptions
Allocation to Configuration Items

System/Software Design
Class Diagrams
Interface Specifications
Algorithms
Form Factor

Implementation/Coding
Integration and Test
Sequence Diagrams
Qualification Testing

ICSM SysML Functional Model

ibd [Block] ICSM [ICSM Blocks]

System Boundary

«block» Power Bus

«block» Baseband Control Bus

«block» PowerSystem

«block» SystemControl

«block» BIT

«block» ADC / DAC

«block» PlatformDynamics

«block» Network

«block» SignalProcessing

«block» SecurityService

«block» RF

«block» Antenna

«block» Baseband Data Bus

«block» UserIF

0..*
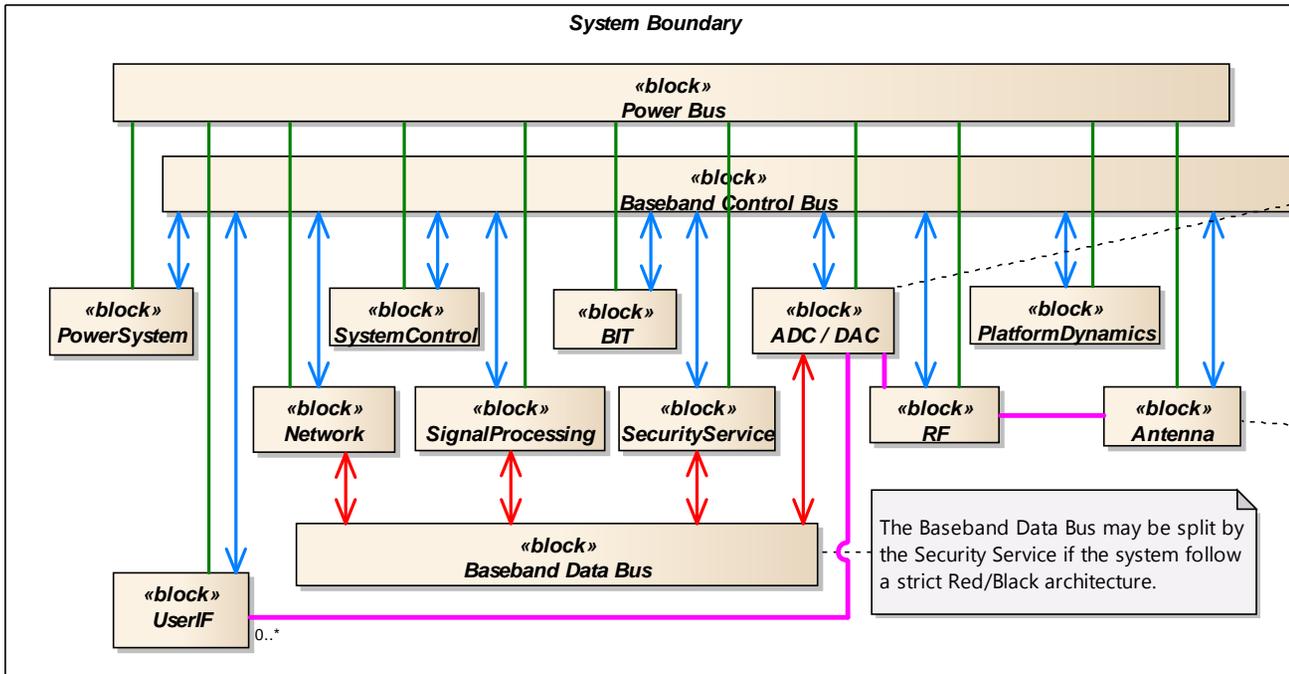
For systems with a voice input/output for the user, there will typically be a CODEC associated with the user interface. Also, if the system is split into a Red/Black side by the SecurityService, then there will be ADC / DAC module on both sides.

May be more than one antenna in large systems. If so, then the RF block will require the capability to selectively route the analog RF signal appropriately.

The Baseband Data Bus may be split by the Security Service if the system follow a strict Red/Black architecture.

SysML enables a system engineering view

Each element has data and information attached to the element.

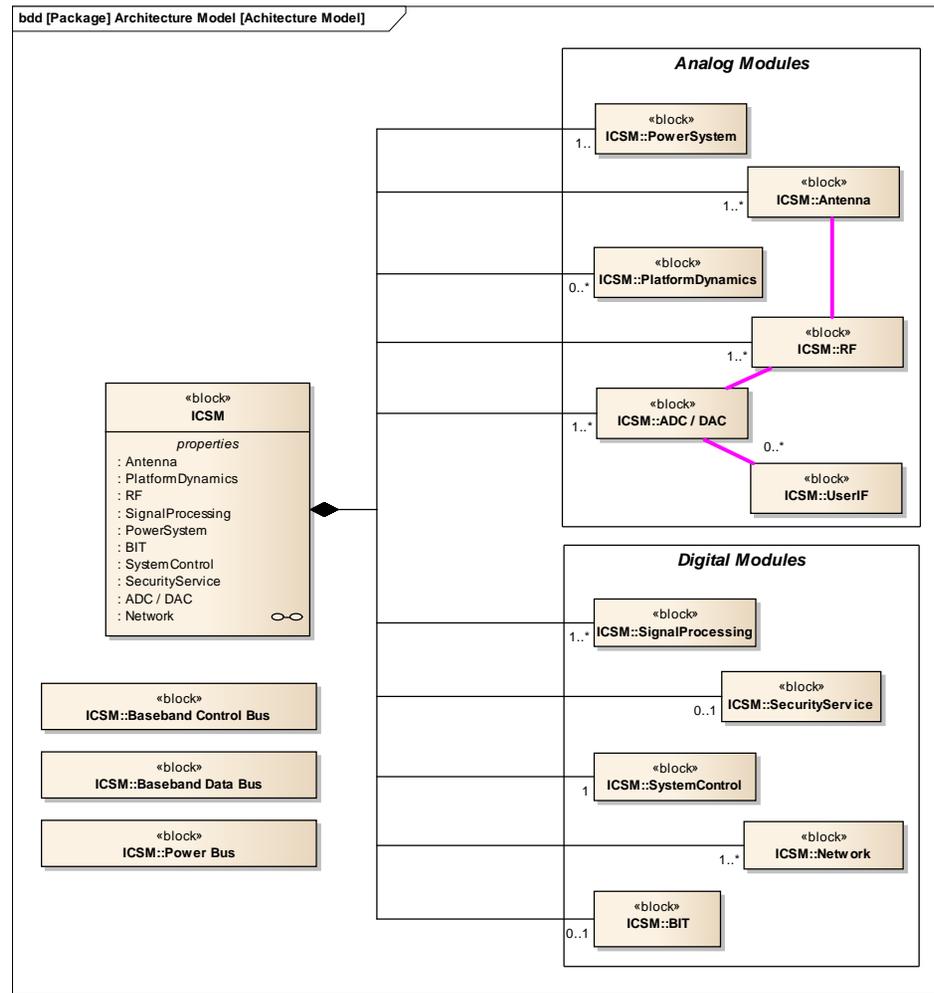**This promotes multiple views but a single model of the system.**

ICSM Functional Decomposition

Driving the future of radio communications and systems worldwide

ICSM Use Case Organization

uc [Package] Use Cases [Use Cases]

System Startup and Shutdown Use Cases and Scenarios

Start System
- + SystemStart
- + Start Antenna
- + Start BIT
- + Start Digital Signal Processor
- + Start Platform Dynamics
- + Start Power Module
- + Start RF
- + Start Security Services
- + Start System
- + Start System Control

Stop System
- + SystemStop
- + Stop Antenna
- + Stop Bit
- + Stop Digital Signal Processor
- + Stop Platform Dynamics
- + Stop Power Module
- + Stop RF
- + Stop Security Service
- + Stop System
- + Stop System Control

Operational Use Cases and Scenarios

System Control

Resource Management

BIT

Load Application

Teardown Application

Maintenance Use Cases and Scenarios
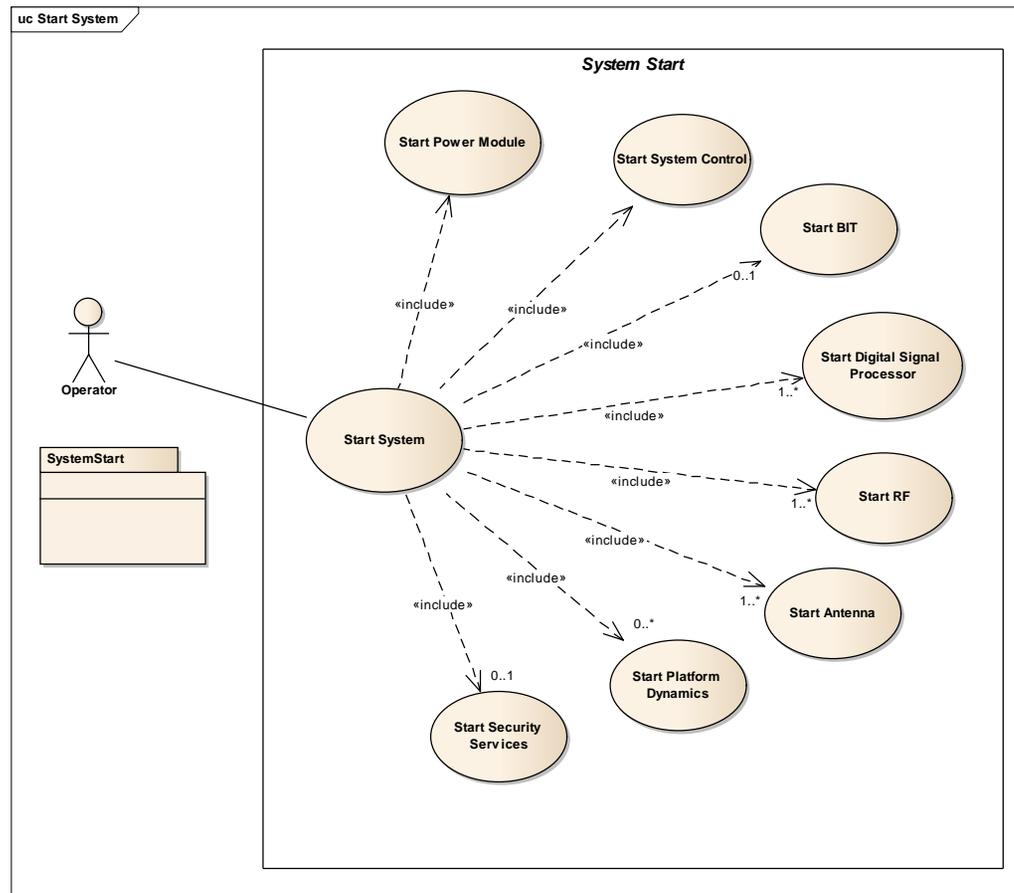
Manage Security

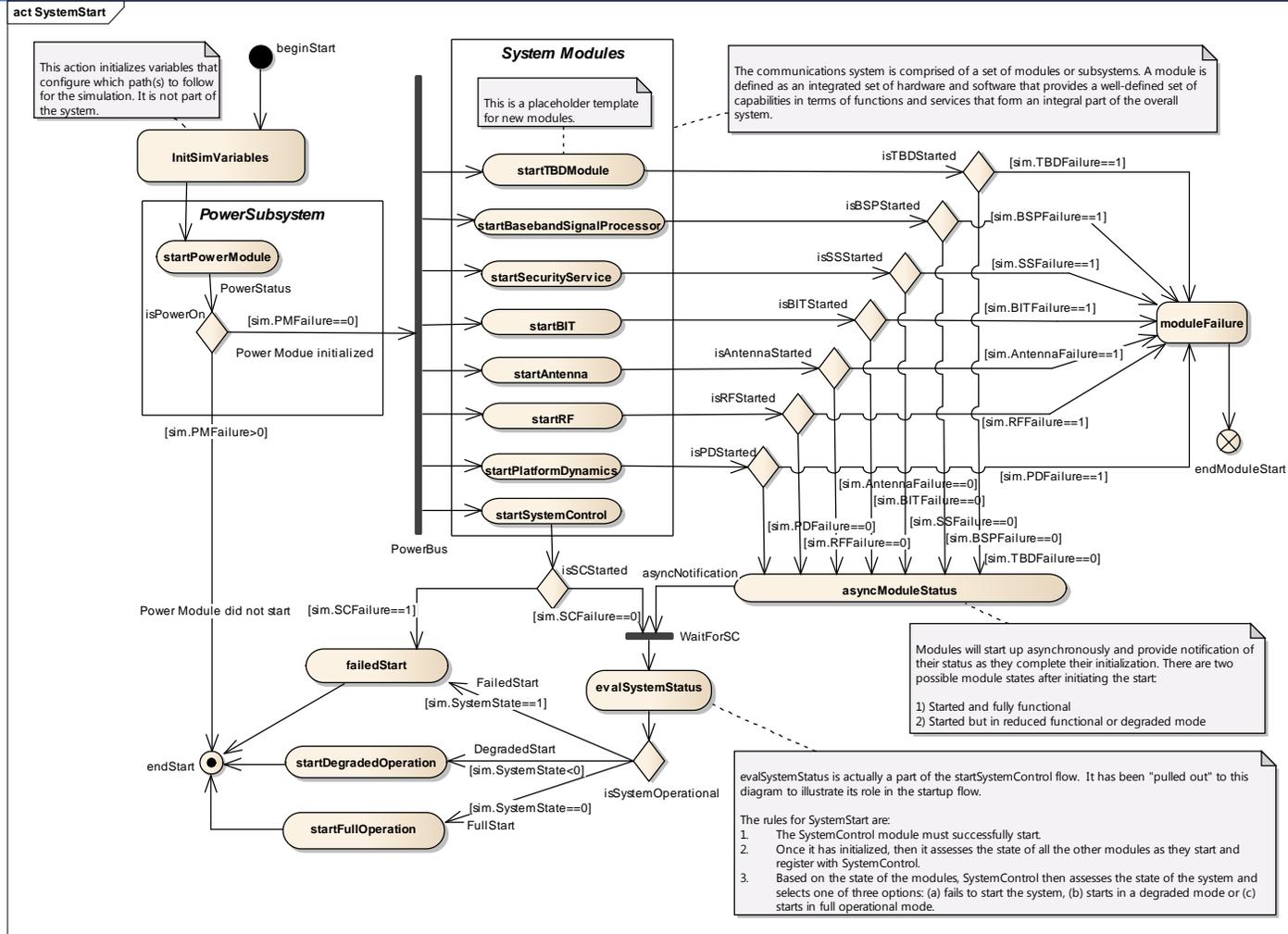**Integrate Module**

Install Application
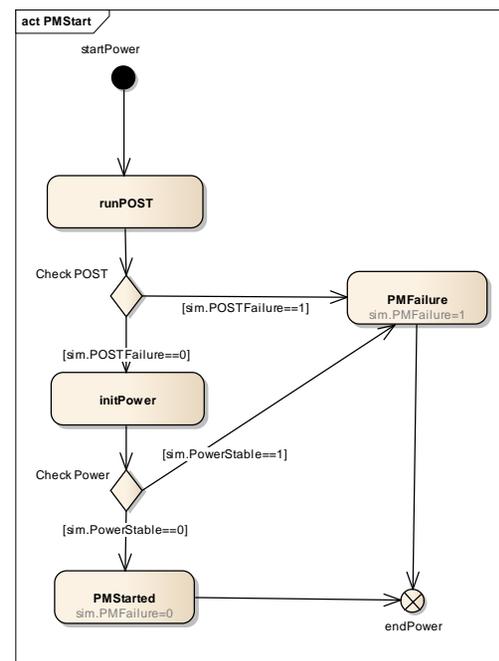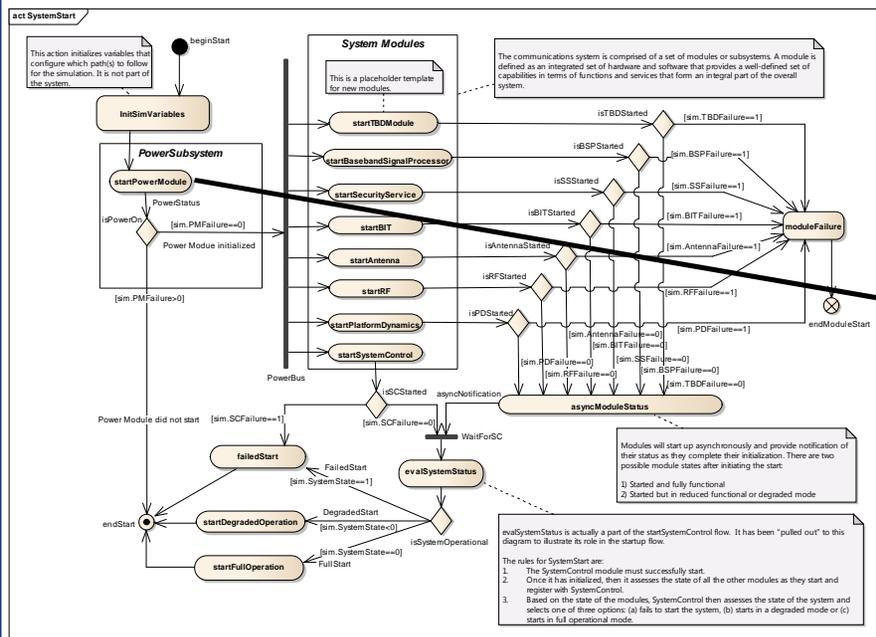
Remove Application

Start System Use Case

System Start Activity Diagram

act SystemStart

beginStart

This action initializes variables that configure which path(s) to follow for the simulation. It is not part of the system.

InitSimVariables

PowerSubsystem

startPowerModule

PowerStatus

isPowerOn

[sim.PMFailure==0]

Power Modue initialized

[sim.PMFailure>0]

PowerBus

System Modules

This is a placeholder template for new modules.

The communications system is comprised of a set of modules or subsystems. A module is defined as an integrated set of hardware and software that provides a well-defined set of capabilities in terms of functions and services that form an integral part of the overall system.

startTBDModule — isTBDStarted — [sim.TBDFailure==1]

startBasebandSignalProcessor — isBSPStarted — [sim.BSPFailure==1]

startSecurityService — isSSStarted — [sim.SSFailure==1]

startBIT — isBITStarted — [sim.BITFailure==1]

startAntenna — isAntennaStarted — [sim.AntennaFailure==1]

startRF — isRFStarted — [sim.RFFailure==1]

startPlatformDynamics — isPDStarted — [sim.PDFailure==1]

startSystemControl

moduleFailure

endModuleStart

[sim.AntennaFailure==0]
[sim.BITFailure==0]
[sim.PDFailure==0]   [sim.SSFailure==0]
       [sim.RFFailure==0]   [sim.BSPFailure==0]
                                        [sim.TBDFailure==0]

isSCStarted   asyncNotification

asyncModuleStatus

Modules will start up asynchronously and provide notification of their status as they complete their initialization. There are two possible module states after initiating the start:

1) Started and fully functional
2) Started but in reduced functional or degraded mode

Power Module did not start   [sim.SCFailure==1]   [sim.SCFailure==0]   WaitForSC

failedStart

FailedStart

[sim.SystemState==1]

endStart

evalSystemStatus

evalSystemStatus is actually a part of the startSystemControl flow. It has been "pulled out" to this diagram to illustrate its role in the startup flow.

The rules for SystemStart are:
1. The SystemControl module must successfully start.
2. Once it has initialized, then it assesses the state of all the other modules as they start and register with SystemControl.
3. Based on the state of the modules, SystemControl then assesses the state of the system and selects one of three options: (a) fails to start the system, (b) starts in a degraded mode or (c) starts in full operational mode.

startDegradedOperation

DegradedStart

[sim.SystemState<0]

isSystemOperational

startFullOperation

[sim.SystemState==0]

FullStart

Activity Decomposition

Each activity can be decomposed to lower level detail

This allows incremental development of use cases to validate operational scenarios and requirements

Status and Plans

- **A database repository for the model has been tested and validated.**

- **Current work on the use cases will continue with more detail on the startup and initial development of the shutdown.**

- **Development of reference models for current systems planned:**

  - Zedboard - Xlinx Zynq7000 with RF

  - Ettus Research – USRP N200