High-Assurance Cyber Military Systems

AADL Standards Committee
7 June 2017

Darren Cofer
Trusted Systems
Advanced Technology Center
darren.cofer@rockwellcollins.com

Rockwell Collins

© Copyright 2016 Rockwell Collins, Inc.
All rights reserved.

The Dream…

© Copyright 2017 Rockwell Collins, Inc.
All rights reserved.

High-Assurance Cyber Military Systems (HACMS)

Architectural-Level
Rockwell Collins, University of Minnesota
Compositional Reasoning

Application-Level Software
Galois, CMU, Draper Labs, MIT, Oxford, Princeton, SpiralGen, University of Illinois, University of Pennsylvania
Generate from Specification, Correct by Construction, Software Verification, Robust Algorithms

Low-Level Software
Data61, Yale
Verified OS Kernels

Ground Vehicle
HRL
Integrate on TARDEC Autonomous Systems

Air Vehicle
Boeing
Integrate on Unmanned Little Bird

DARPA

The HACMS program is creating technology for the construction of *high-assurance* cyber-physical systems

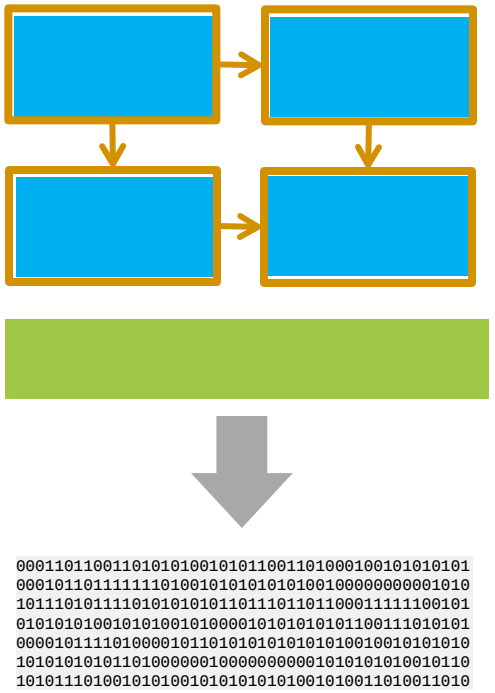
Use *formal methods* to build systems that are resilient against cyber-attack because they can be proven not to have typical security vulnerabilities

Evaluation & Penetration Testing
Draper, AIS

© Copyright 2017 Rockwell Collins, Inc.
All rights reserved.

Goal: Architecture-Driven Assurance

- Architecture model is correct
  - Properties, structure, behavior, interaction of components, interfaces, contracts
  - Analyzable
- Components are correct
  - Consistent/realizable contracts
  - Components verified to implement contracts
- System does what the model says
  - No other information flows (memory safety, isolation)
  - OS executes model correctly (incl. timing)
- System implementation corresponds to model
  - Automatic build from component and architecture models



© Copyright 2017 Rockwell Collins, Inc.
All rights reserved.

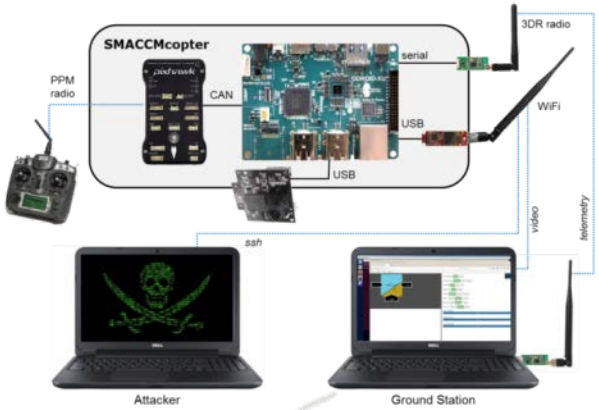
HACMS Accomplishments: Technologies

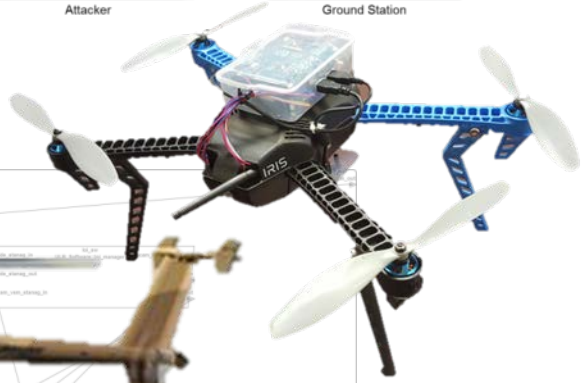
- **Open source tools, languages, software**
- seL4 formally verified kernel
  - New functionality
  - Verification for 64-bit/x86
- Ivory/Tower embedded DSLs
  - Memory safe component software
- Architecture modeling and analysis tools (AADL)
  - Assume-Guarantee Reasoning Environment (AGREE)
  - Architecture-based assurance cases (Resolute)
  - Trusted Build from AADL model, verified components, verified OS

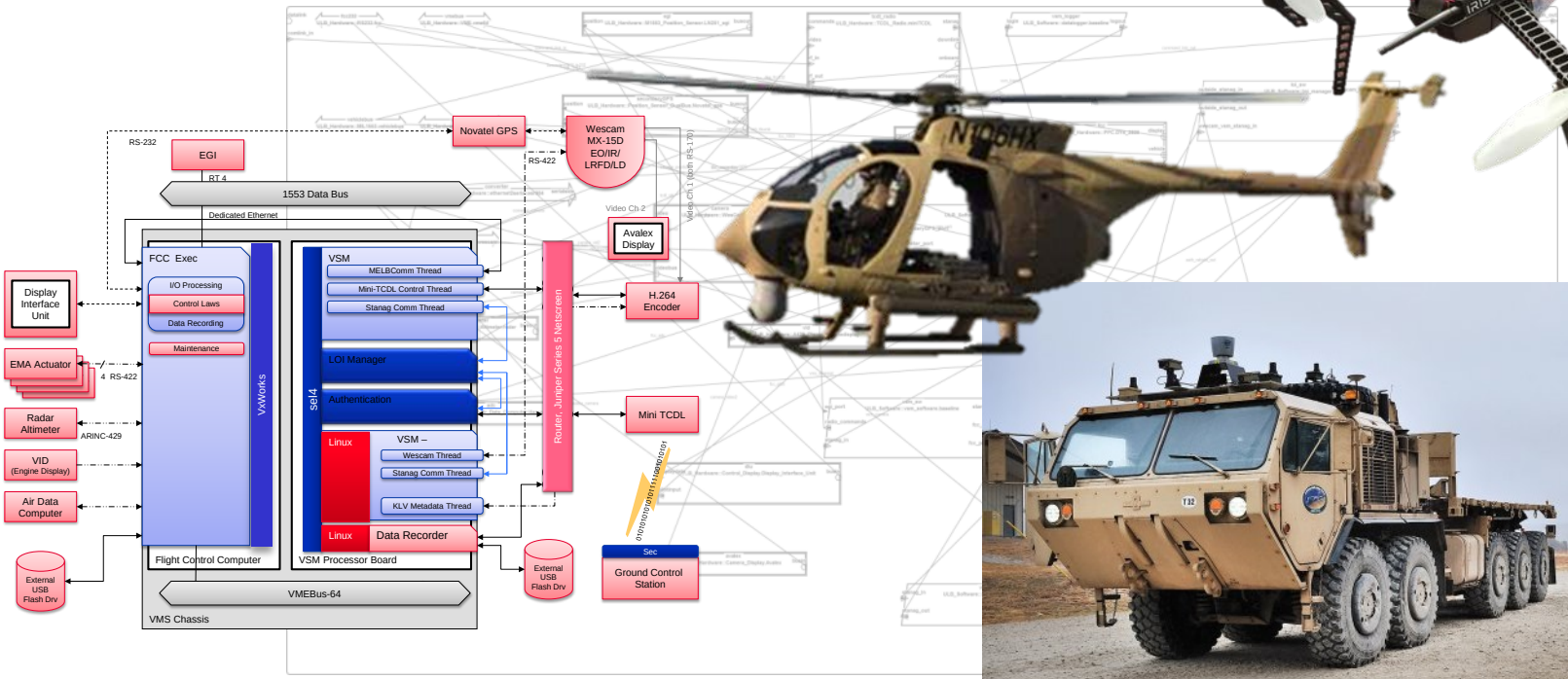
© Copyright 2017 Rockwell Collins, Inc.
All rights reserved.

HACMS Demonstrations

- **Practical and effective**
- Quadcopter
- Boeing Unmanned Little Bird helicopter
- Army TARDEC autonomous HET



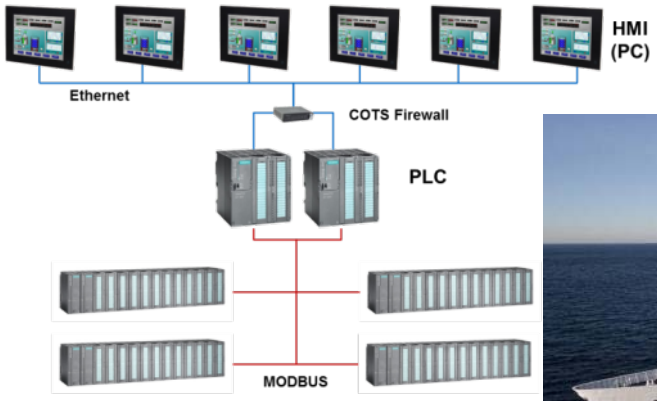




© Copyright 2017 Rockwell Collins, Inc.
All rights reserved.

HACMS Transitions

- Army JMR/FVL
- Navy PIMCS
- More to come…



NAVSEA
NAVAL SEA SYSTEMS COMMAND

AMRDEC

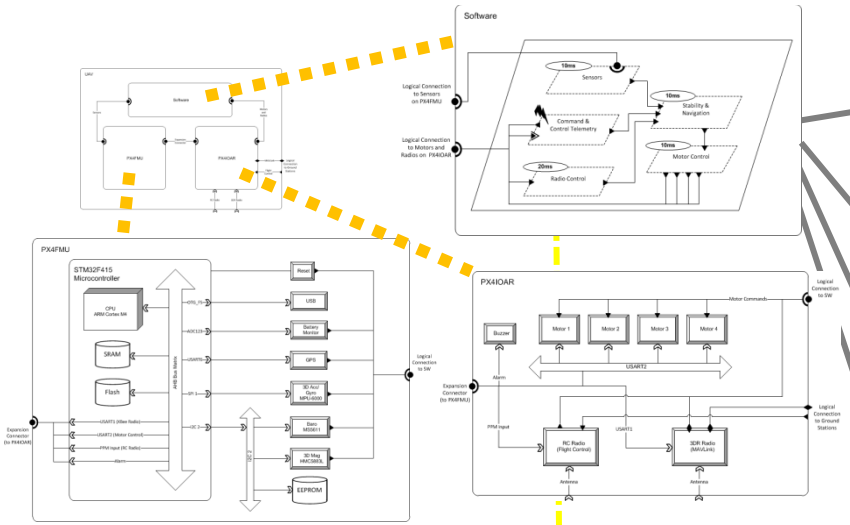
Why Trusted Build?

- Ensure fidelity between models and system image
  - Proofs are over architectural models
    - Information flow between processes and threads
    - Well-formedness of architecture: scheduling, memory limits and safety, etc.
  - Trusted build *generates* system image from architectural model
    - Prevents stupid errors
      - Mismatches on unit types between modules [Mars Polar Lander]
      - Mismatches on alignment of data, data representation, and data location

Credit: Mike Whalen, UMN

© Copyright 2017 Rockwell Collins, Inc.
All rights reserved.

CAmkES / seL4 Build process



Directories containing C code

CAmkES Assembly

```
assembly {
composition {
  component dispatch_periodic
    dispatch_periodic_inst;
…
```

CAmkES Component

CAmkES Makefile

```
TARGETS := $(notdir ${SOURCE_DIR}).cdl
ADL := test1_impl_assembly.camkes
TEMPLATES := ../../projects/templates
../../projects/global-components/templates
…
```

CAmkES Config

```
component receiver {
  include "smaccm_test4_impl_types.h";
  control;
    provides uint32_t_writer Input1;
    …
```

AADL C Glue Code

```
void
Test_AADL__Thread_A_main(
        bool signalCh) {
    init_thread_A();
    …
```

Component C Code

```
void
exec_sig_systick(void) {
systick_signal_write_chEmitter();
}
…
```

CAmkES make

Image

00001101000100…

Grey Arrows: Trusted Build
Dark Blue Arrows: CAmkES
Yellow: hand-written code/models

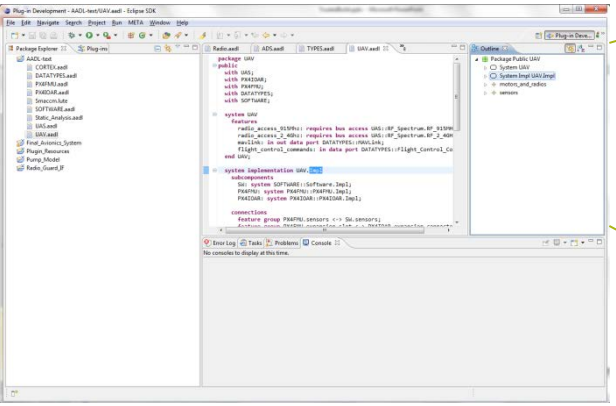
© Copyright 2017 Rockwell Collins, Inc.
All rights reserved.

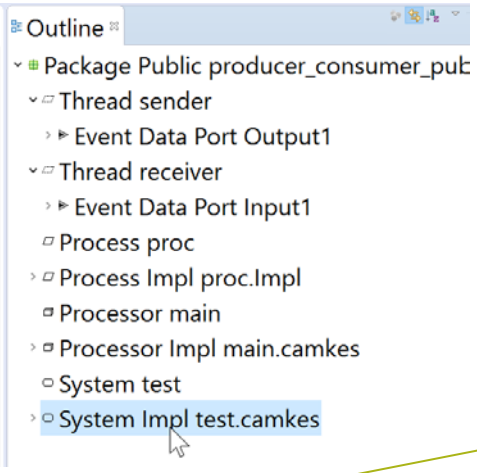
Why *Trusted* Build?

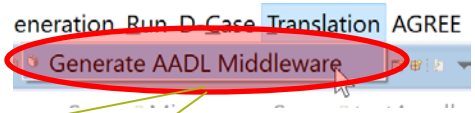
- OSATE plug-in for Eclipse
- Written in Java using String Template library
- No high-assurance pedigree, except:
  - Implementations of some connectors (Mailboxes for Dataports (seL4/Linux), Connector components for data/event ports (seL4) have associated proofs
  - Code has been inspected by multiple groups, including Red Team, looking for security flaws
- Eventual goal is to tie into CAmkES/seL4 connector proofs
  - Isabelle/HOL

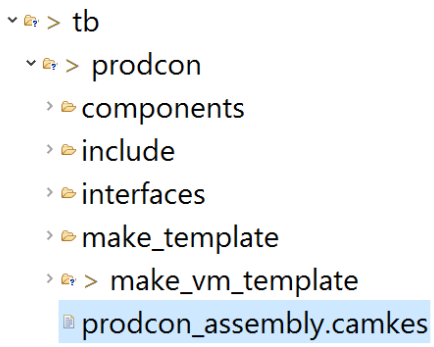
© Copyright 2017 Rockwell Collins, Inc.
All rights reserved.

Generating Code:









Components/IDL Files

```
…
assembly {
   composition {
      component dispatch_periodic dispatch_periodic_inst;

      component TimeServerKZM time_server;
      // Component instances for all AADL-defined threads
      component sender sender_inst;
      component receiver receiver_inst;

      // Port declarations for active threads
      connection seL4TimeServer tb_sender_periodic_dispatcher_timer(from
sender_inst.tb_timer, to time_server.the_timer);
      connection seL4Notification tb_sender_periodic_dispatcher_echo_int(from
dispatch_periodic_inst.sender_periodic_dispatcher, to sender_inst.tb_timer_complete);
```

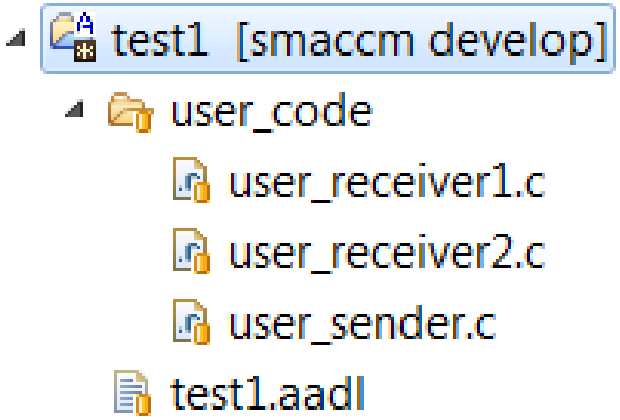
© Copyright 2017 Rockwell Collins, Inc.
All rights reserved.

Structure of Generated Code

- CAmkES OS Configuration
  - .camkes component files
  - CAmkES assembly file for top-level system
  - IDL files describing RPCs
- Generated C Middleware Code
  - tb_<component>.c
    - Implementation of middleware services
  - tb_<component>.h
    - Description of API between user-level AADL code and TB code.
- Template Makefile for system build
  - Works for simple build procedures
  - However, for more complex system generations, likely you will want to replace it with a more “full-featured” make

© Copyright 2017 Rockwell Collins, Inc.
All rights reserved.

Simple Model Initial Structure

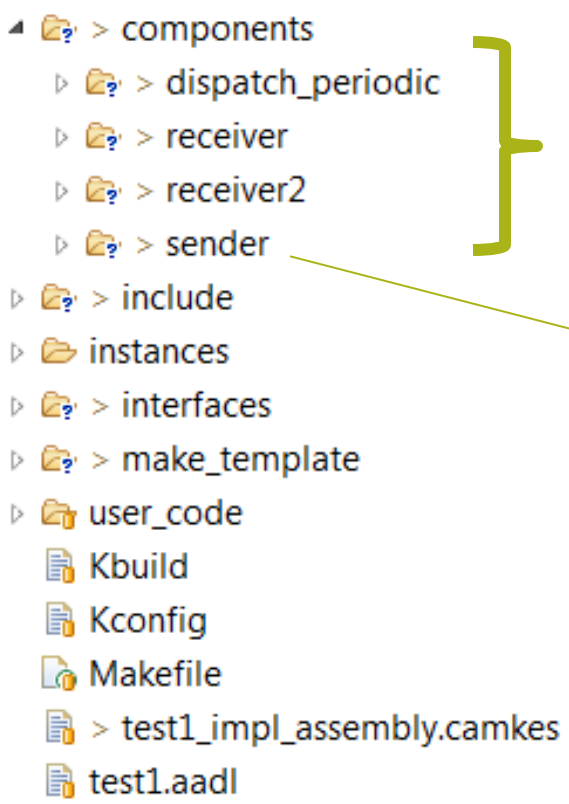


C files describe the component behavior; in this case, there is one C file per component.

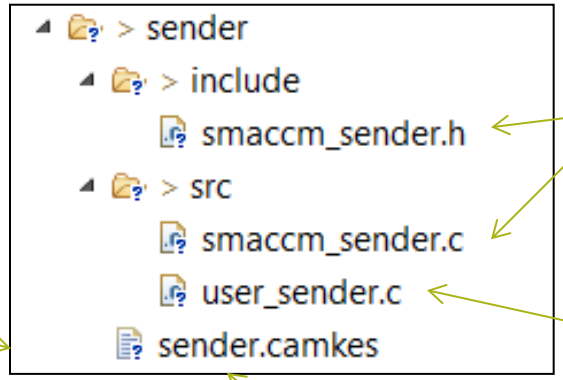
AADL files describe the system structure

© Copyright 2017 Rockwell Collins, Inc.
All rights reserved.

Structure of Generated Code



Each AADL thread becomes a CAmkES component



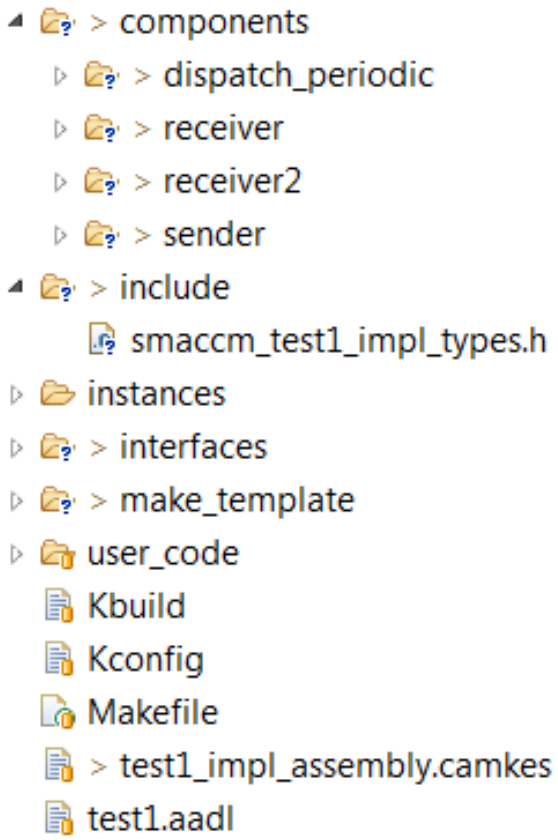
Autogenerated files for the middleware implementation

User-provided code for component.

CAmkES interface file

© Copyright 2017 Rockwell Collins, Inc.
All rights reserved.

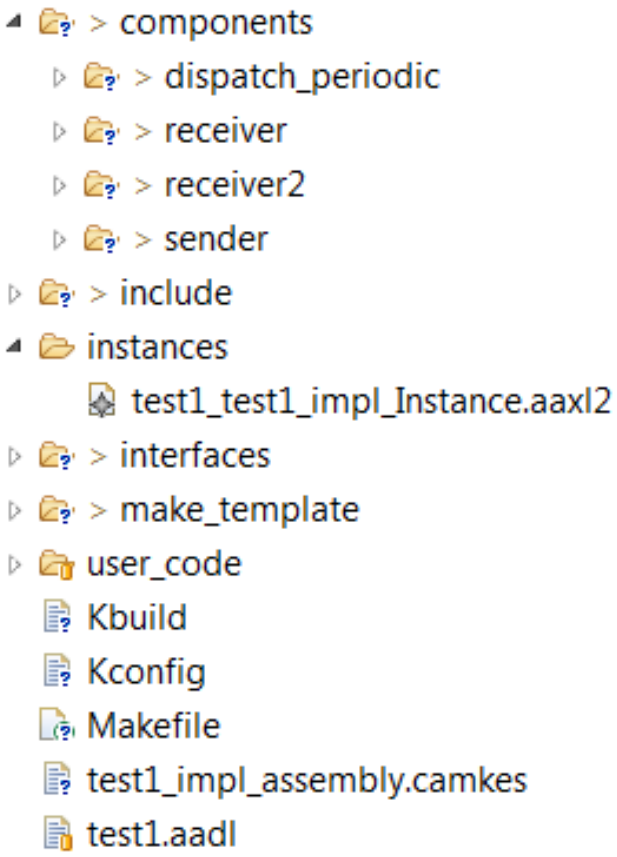
Structure of Generated Code



Common include files for all components. In this case, only the AADL type definitions.

© Copyright 2017 Rockwell Collins, Inc.
All rights reserved.

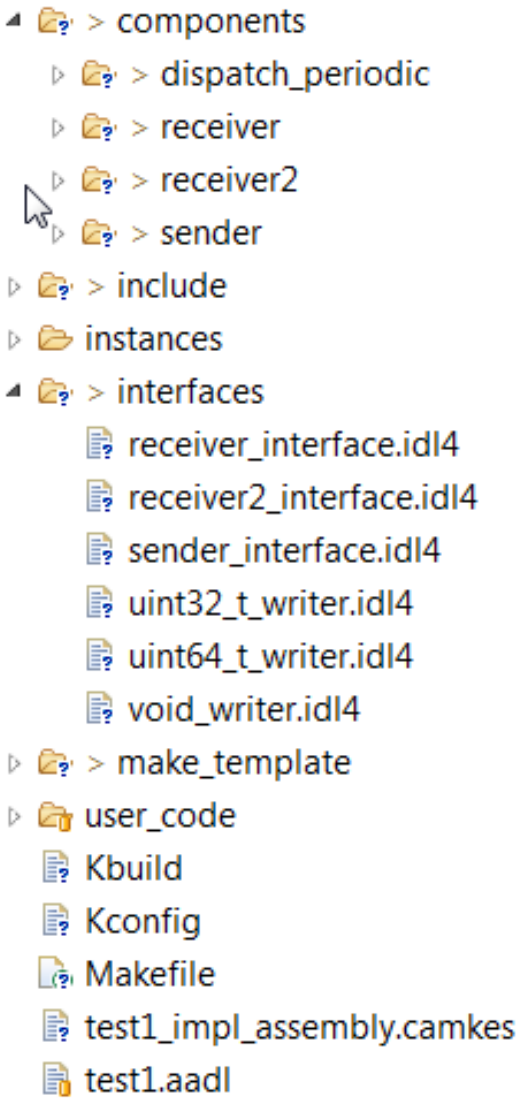
Structure of Generated Code



OSATE/AADL-generated XML file for the system instance. This file can be ignored.

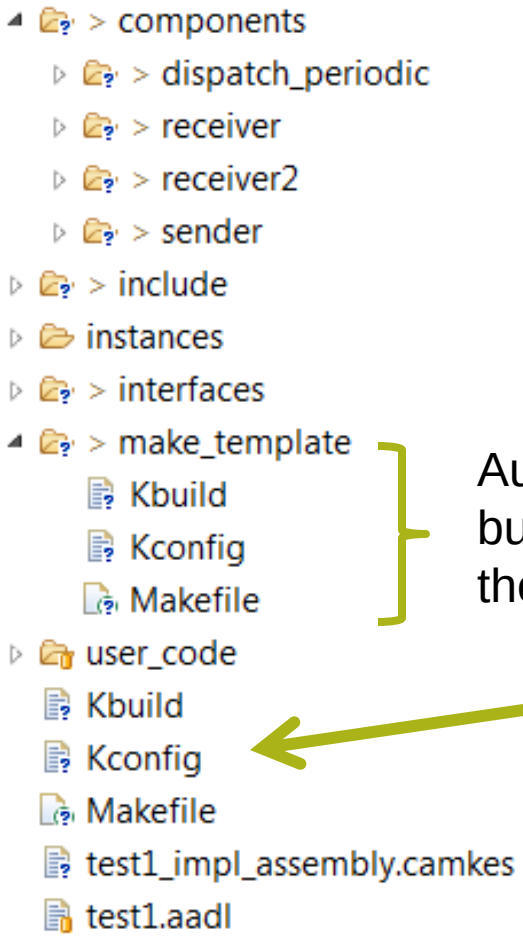
© Copyright 2017 Rockwell Collins, Inc.
All rights reserved.

Structure of Generated Code



Interface definitions for components.

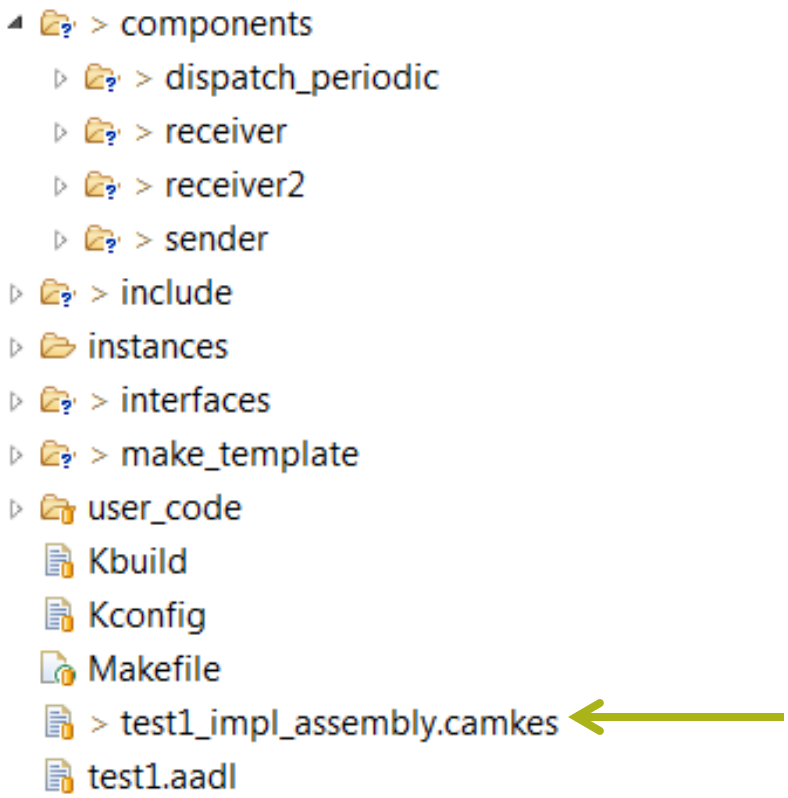
Structure of Generated Code



Autogenerated templates for makefiles and related build files required by CAmkES. To use them, copy them to the parent directory.

© Copyright 2017 Rockwell Collins, Inc.
All rights reserved.

Structure of Generated Code



Top level assembly file for CAmkES ADL.

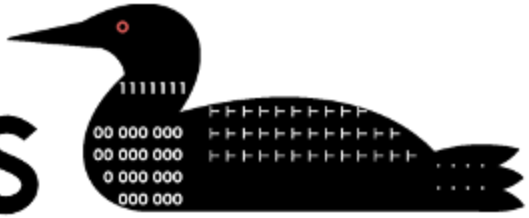
© Copyright 2017 Rockwell Collins, Inc.
All rights reserved.

Wrap Up

- Trusted Build is a tool for generating system implementation skeletons from AADL models
- Goal is to make AADL analysis meaningful against generated code
- It supports a subset of AADL
  - This is intentional: goal is to have small enough code base to have confidence in result
- Can target multiple OS: seL4, VxWorks, eChronos, Linux

© Copyright 2017 Rockwell Collins, Inc.
All rights reserved.

Loonwerks



More information, code, and papers available at:

Loonwerks.com

© Copyright 2017 Rockwell Collins, Inc.
All rights reserved.