PRISMTECH

Performance Critical Middleware

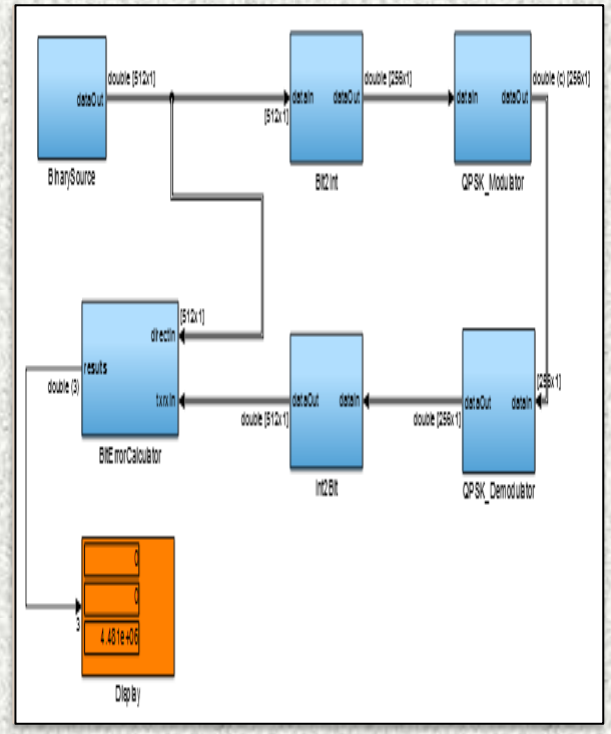
A Model Driven Tool Integration Process for Rapid SDR Development

May 12, 2013

Project Goal

Provide an **integrated** and tool supported waveform development and deployment workflow that **reduces** time to market

Overview of Tools



The MathWorks™ Simulink

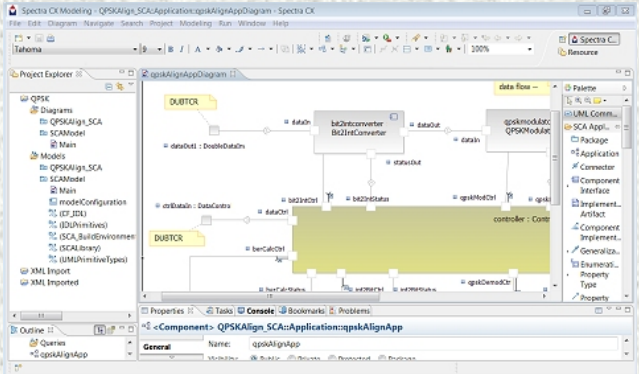
Product Name: Simulink
Version: 2012b

Key Capabilities:

- Model-based design environment
- Simulation and analysis of dynamic systems
- Comprehensive library of blocks
- Code generation

PRISMTECH
Performance Critical Middleware

Overview of Tools



PrismTech Spectra CX

Product Name: Spectra CX

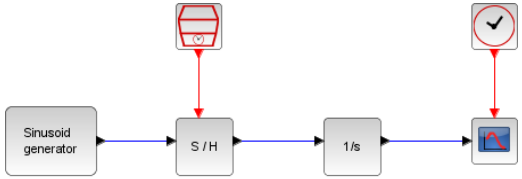
Version: 3.5.1

Key Capabilities:

- Model-driven engineering environment
- Code and test generation for the SCA 2.2.2
- Remote management of platform and applications

PRISMTECH
Performance Critical Middleware

Today's Situation

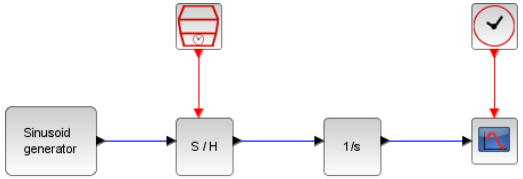


Functional WF Block Model

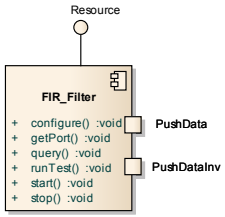
Design Waveform

Simulate & Test Waveform

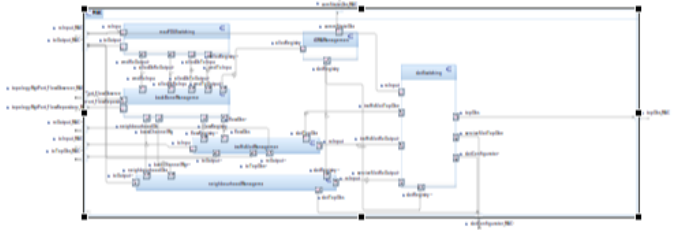
Today's Situation



Functional WF Block Model



Deployable Components



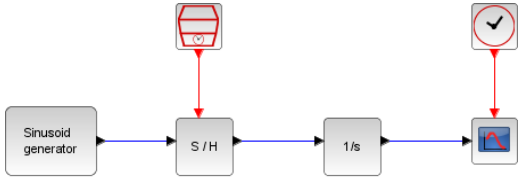
Waveform/Application

Design Waveform Components

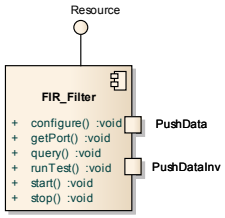
Implement & Test Waveform Components

PRISMTECH
Performance Critical Middleware

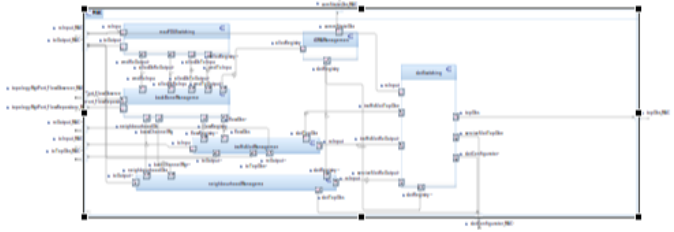
Today's Situation



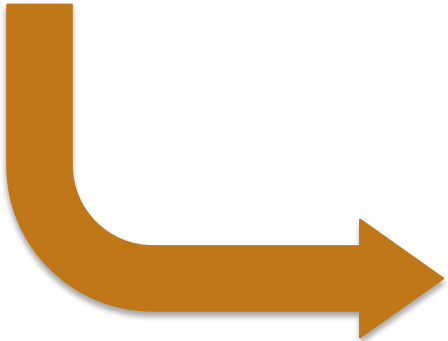
Functional WF Block Model



Deployable Components



Waveform/Application



Deploy Waveform

PRISMTECH
Performance Critical Middleware

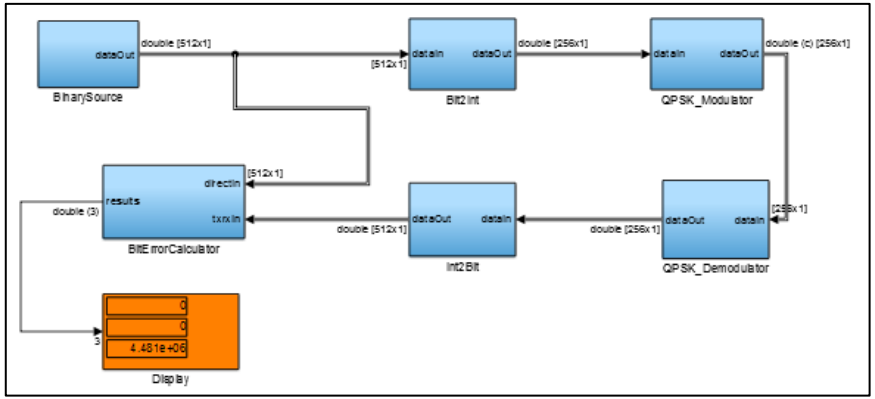
Implications

- Possibility for **errors to be introduced** during hand coding of algorithmic parts
- **Duplication of effort** as algorithm has to be implemented again
- Waveform design not always **in synch** with deployable waveform

A New Workflow…

Design and Simulate Waveform Algorithm

Design and Simulate WF…



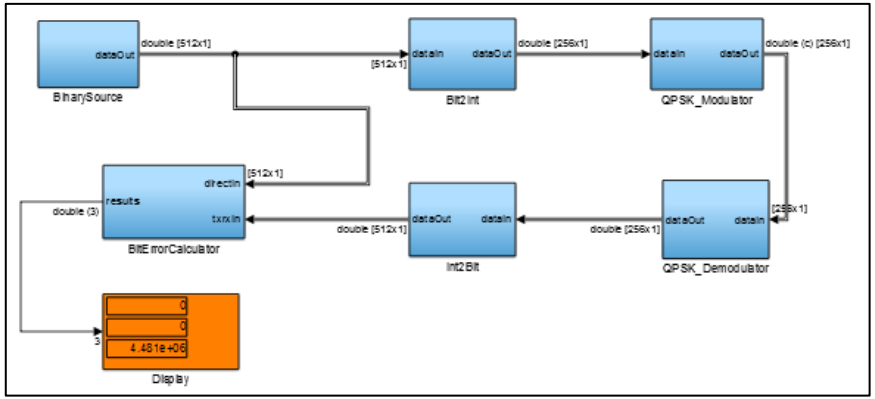
MathWorks Simulink

Design Waveform

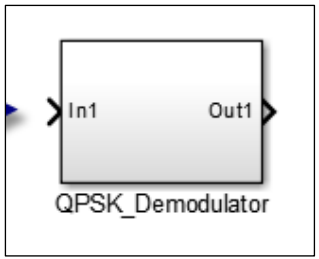
Simulate and Test

PRISMTECH
Performance Critical Middleware

Design and Simulate WF…



MathWorks Simulink



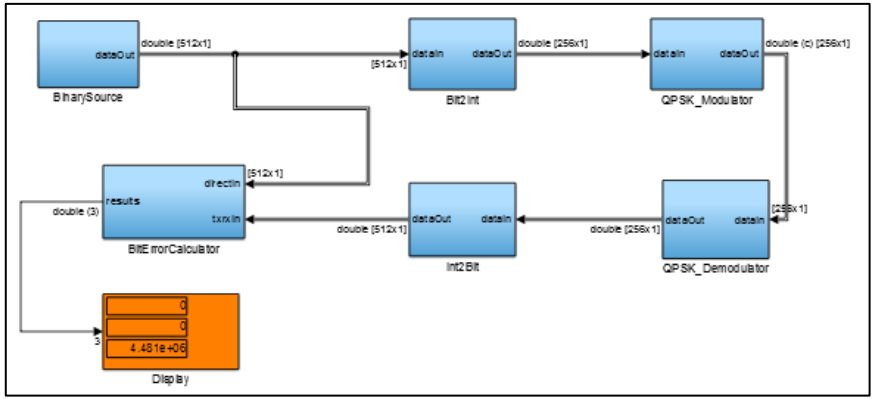
Block

Ensure all blocks are subsystems

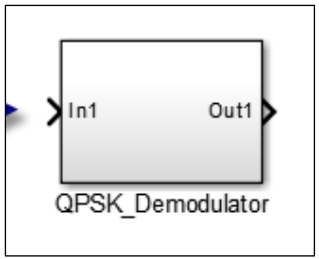
Add platform specifics for RTW

PRISMTECH
Performance Critical Middleware

Design and Simulate WF…



MathWorks Simulink



Block

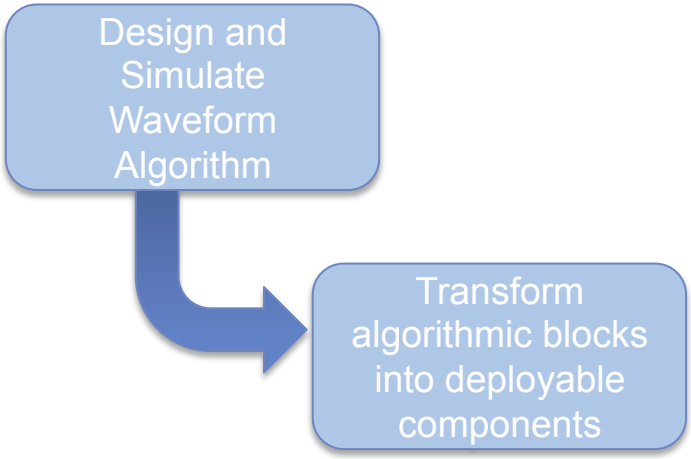
Generate C/C++ for subsystems

- QPSK_Demodulator_private.h
- QPSK_Demodulator_ref.rsp
- QPSK_Demodulator_types.h
- QPSK_Demodulator.cpp
- QPSK_Demodulator.h
- rtmodel.h
- rtw_proj.tmw
- rtwtypes.h

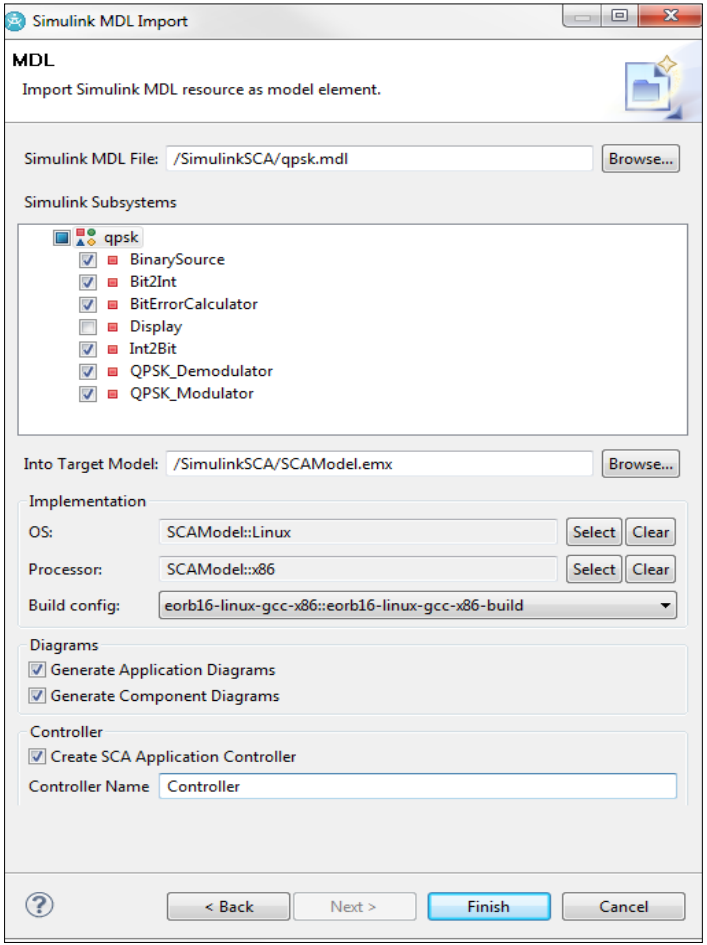
Generated Source Code

PRISMTECH
Performance Critical Middleware

A New Workflow…



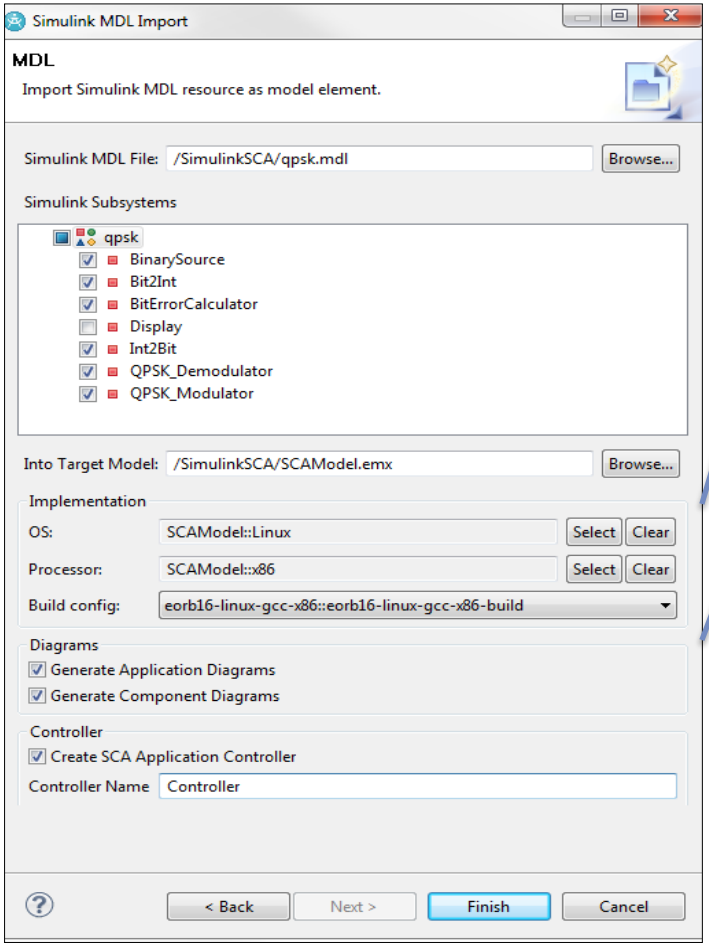
Transform Blocks…

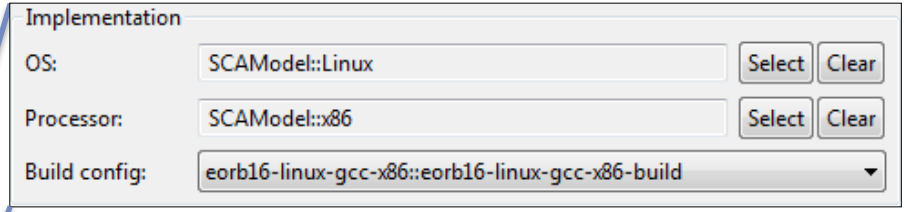


- Wizard guided transformation of design
- Creates a component based design model in CX
- Select which blocks to transform
- Completely automated

PRISMTECH
Performance Critical Middleware

Transform Blocks…



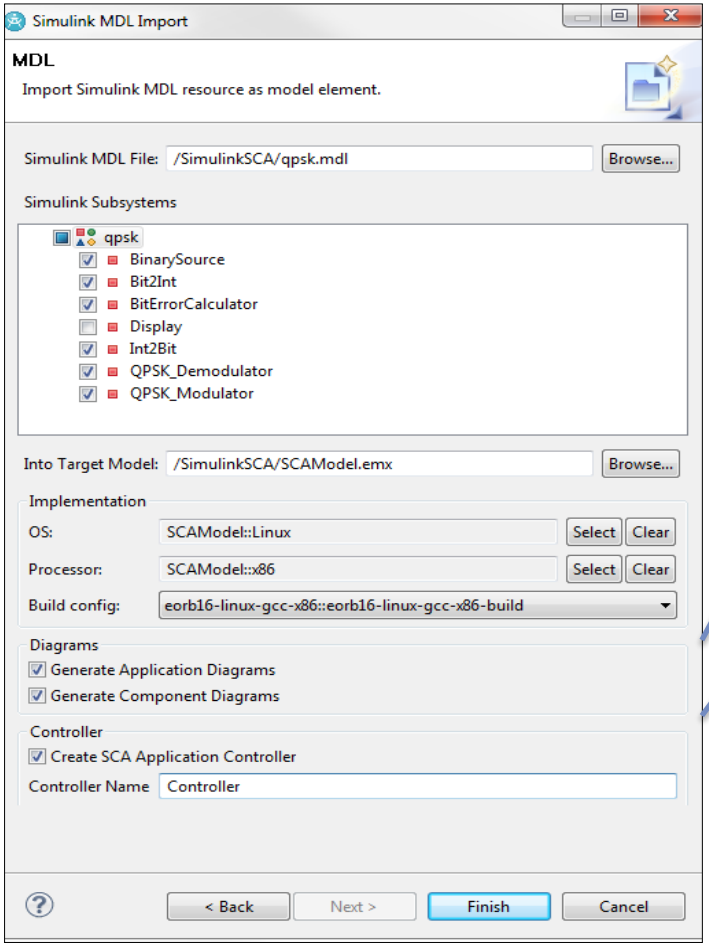


Select target platform for components

Enables CX to generate platform specific code

PRISMTECH
Performance Critical Middleware

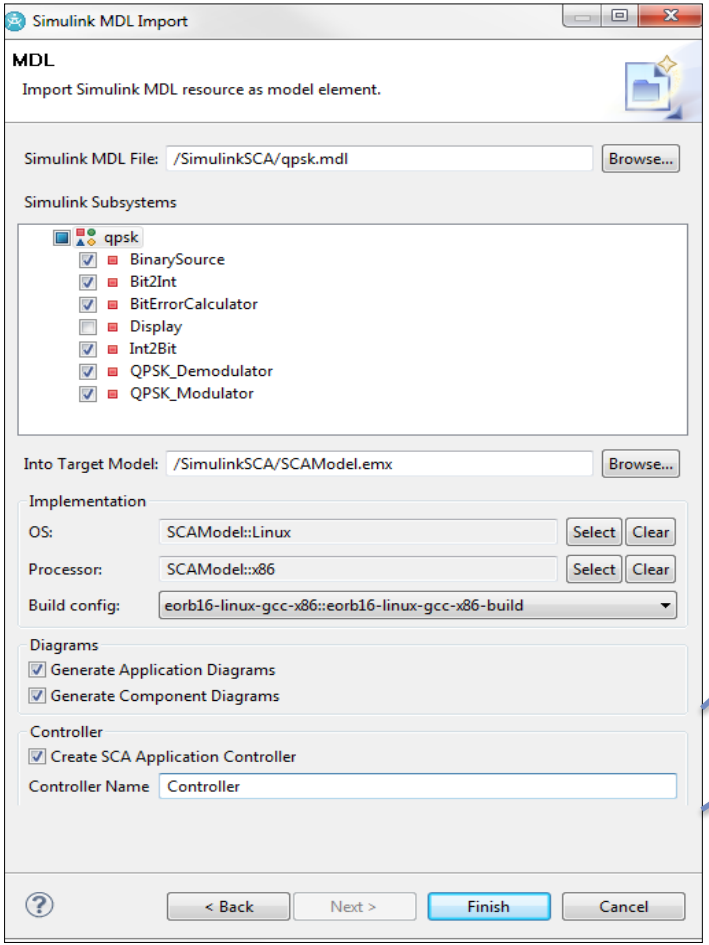
Transform Blocks…



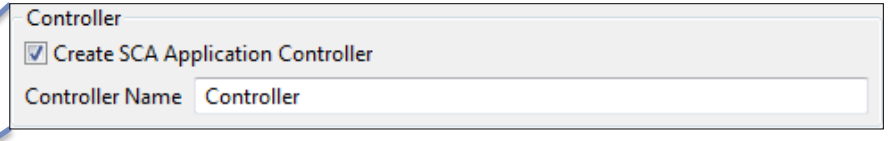
Options to create diagrams as part of transformation

PRISMTECH
Performance Critical Middleware

Transform Blocks…

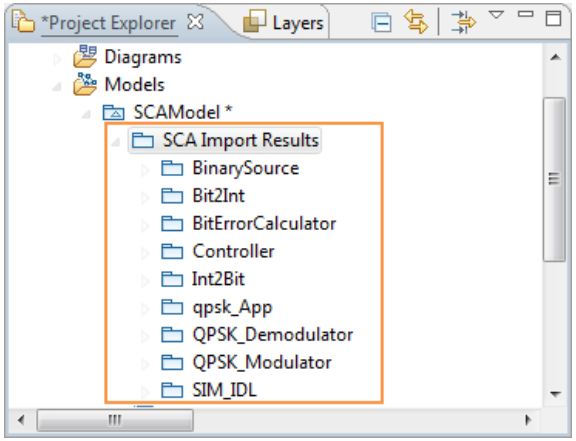


Option to create an application controller

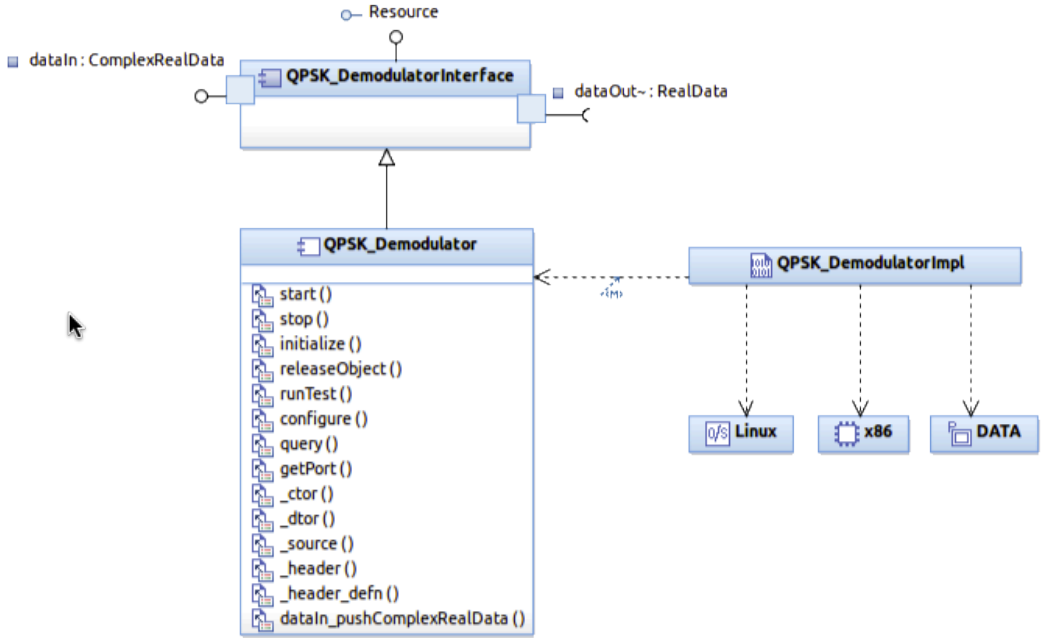


PRISMTECH
Performance Critical Middleware

Components

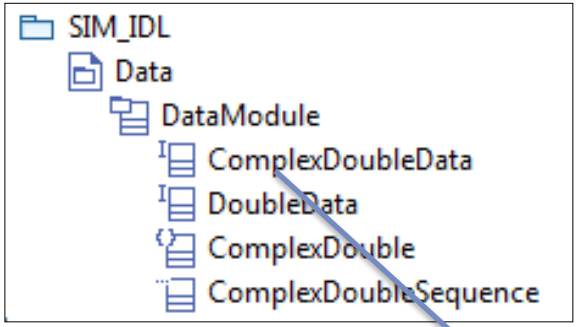


Resulting components can be compiled and deployed



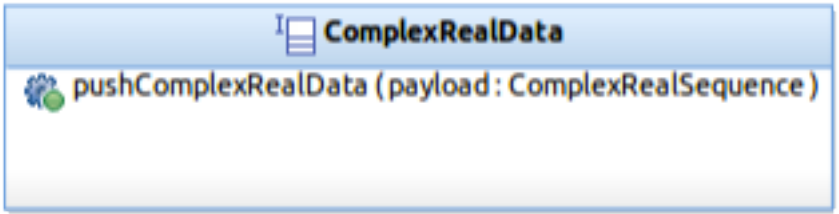
PRISMTECH
Performance Critical Middleware

Interfaces

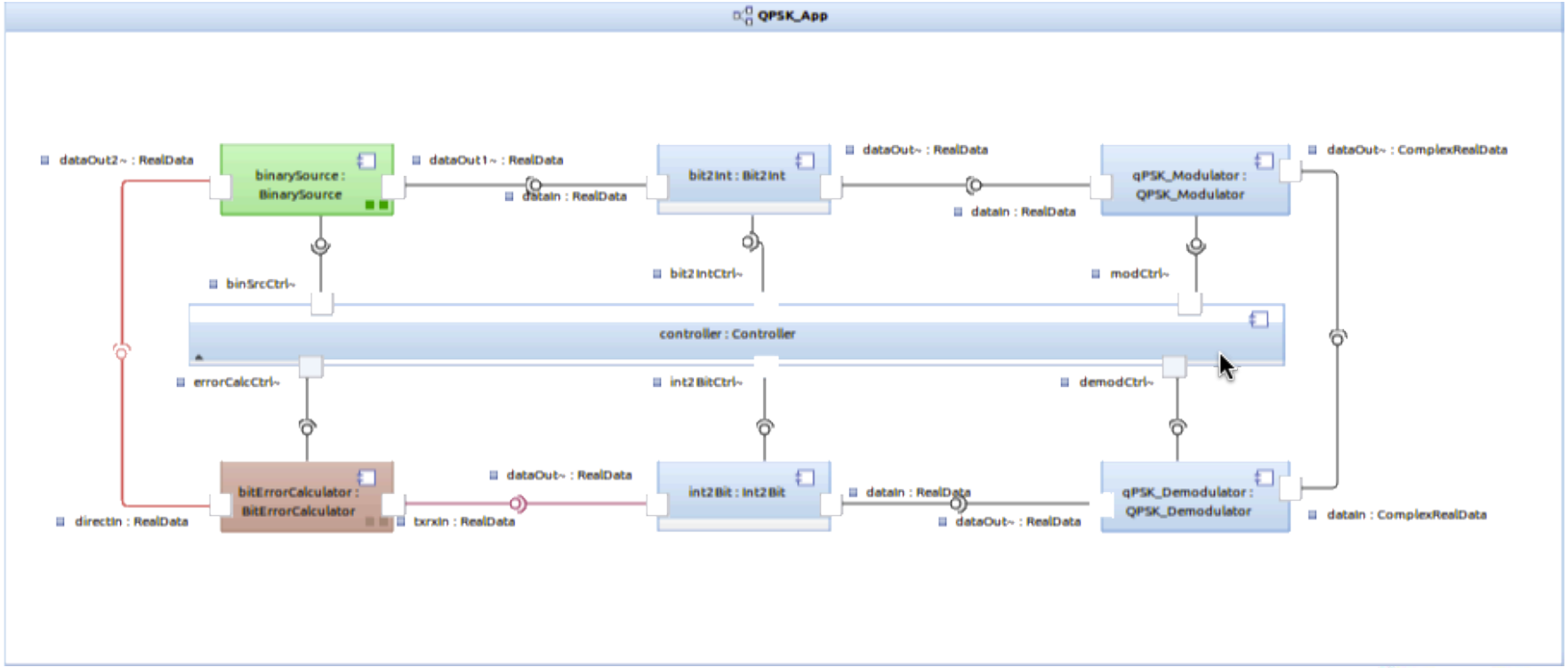


Automated creation of all required interfaces

Interfaces define contracts between components for communication

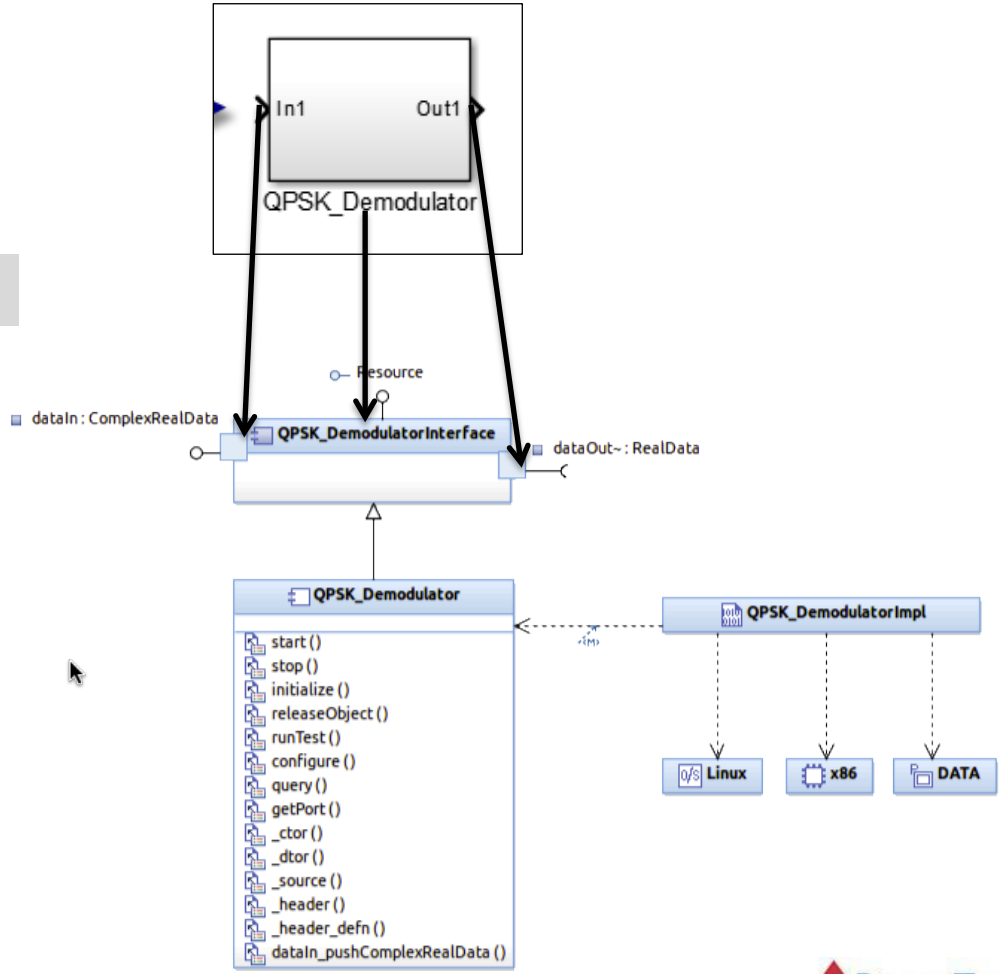


Resulting application is ready to be deployed



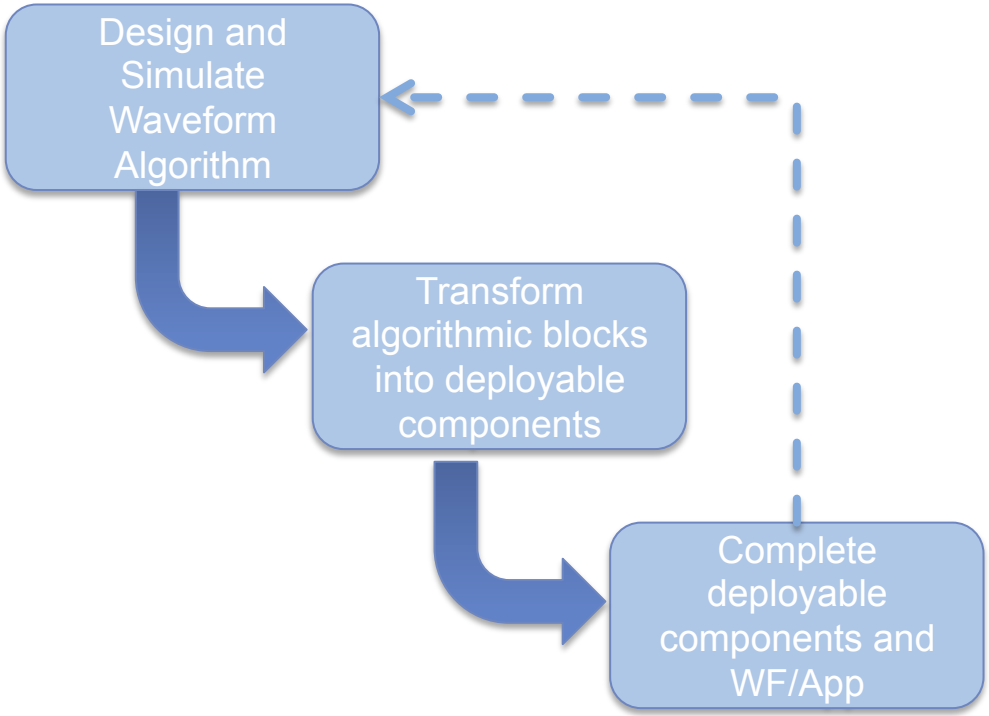
Mapping

Simulink BlockType	Spectra CX
Subsystems	Components
Inport	Provides Ports
Outport	Uses Ports



PRISMTECH
Performance Critical Middleware

A New Workflow…

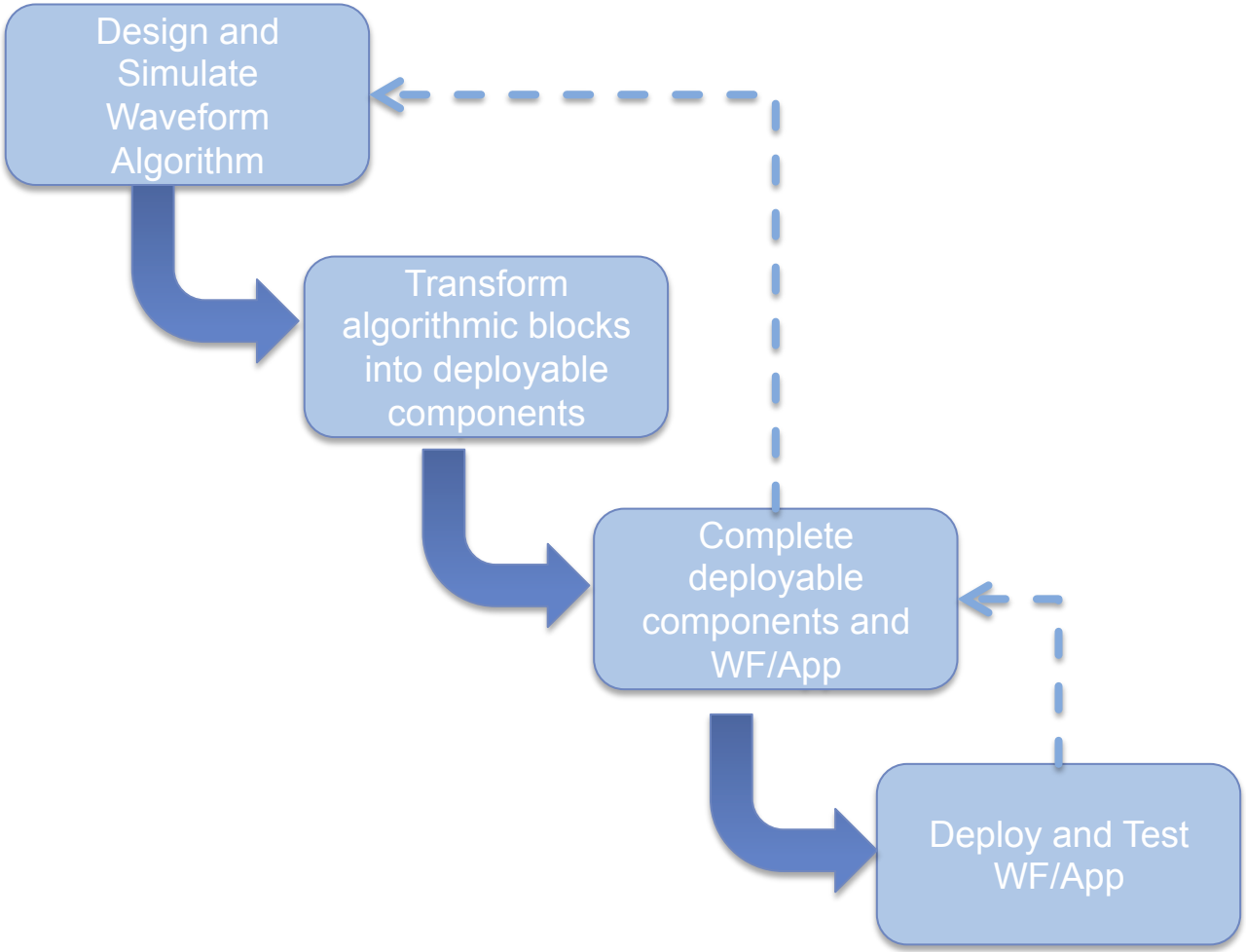


Complete Deployable…

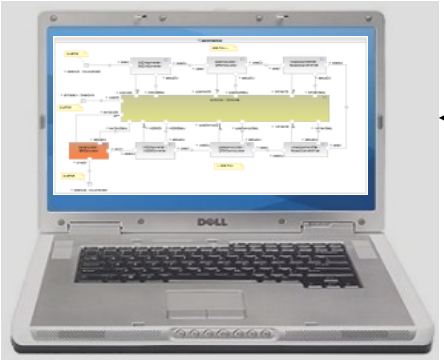
- Add glue code to CX component
- Address timing related issues

```
// fill the input buffer
for(unsigned int x = 0; x < IN_BUFF_SIZE; x++) {
    QPSK_Demodulator_U.dataIn[x].re = payload[x].re;
    QPSK_Demodulator_U.dataIn[x].im = payload[x].im;
}

// step the rtw model
QPSK_Demodulator_step();

// fill the result in the complex output sequence
for(unsigned int x = 0; x < OUT_BUFF_SIZE; x++) {
    dataOutSeq[x] = QPSK_Demodulator_Y.dataOut[x];
}

// send the processed data
dataOut_->pushRealData(dataOutSeq);
```

A New Workflow…



Deploy and Test



Spectra CX

Deploy WF

Instantiate WF

Control WF

Monitor WF

Radio Application (Waveform)

SCA Infrastructure

Radio Application (Waveform)

SCA Infrastructure

Spectra OE: Core Framework

Spectra OE: e*ORB

TCP/IP v4/v6

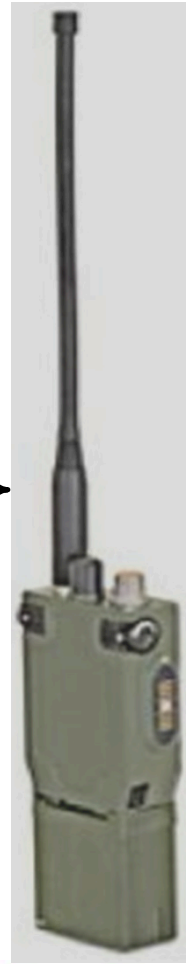
SCA 2.2.2 and POSIX AEP

RTOS / Linux

ASP

BSP

Hardware (GPP, DSP, FPGA)



PRISMTECH

Performance Critical Middleware

Future Work

- Operating Environment in the loop
  - Testing against ‘Golden Waveform’
  - Hand written parts of the waveform
- Automate glue
  - Simulink libraries, object files and header files
  - Behavioral code
  - Optimize glue code

Summary and Benefits

- **Rapid** design, development, testing and deployment
- **Minimize** manual coding effort
- **Reduce** opportunity for human error
- **Reduce** time from requirements to deployment