基于模型开发的面向服务的车载应用软件的设计与测试

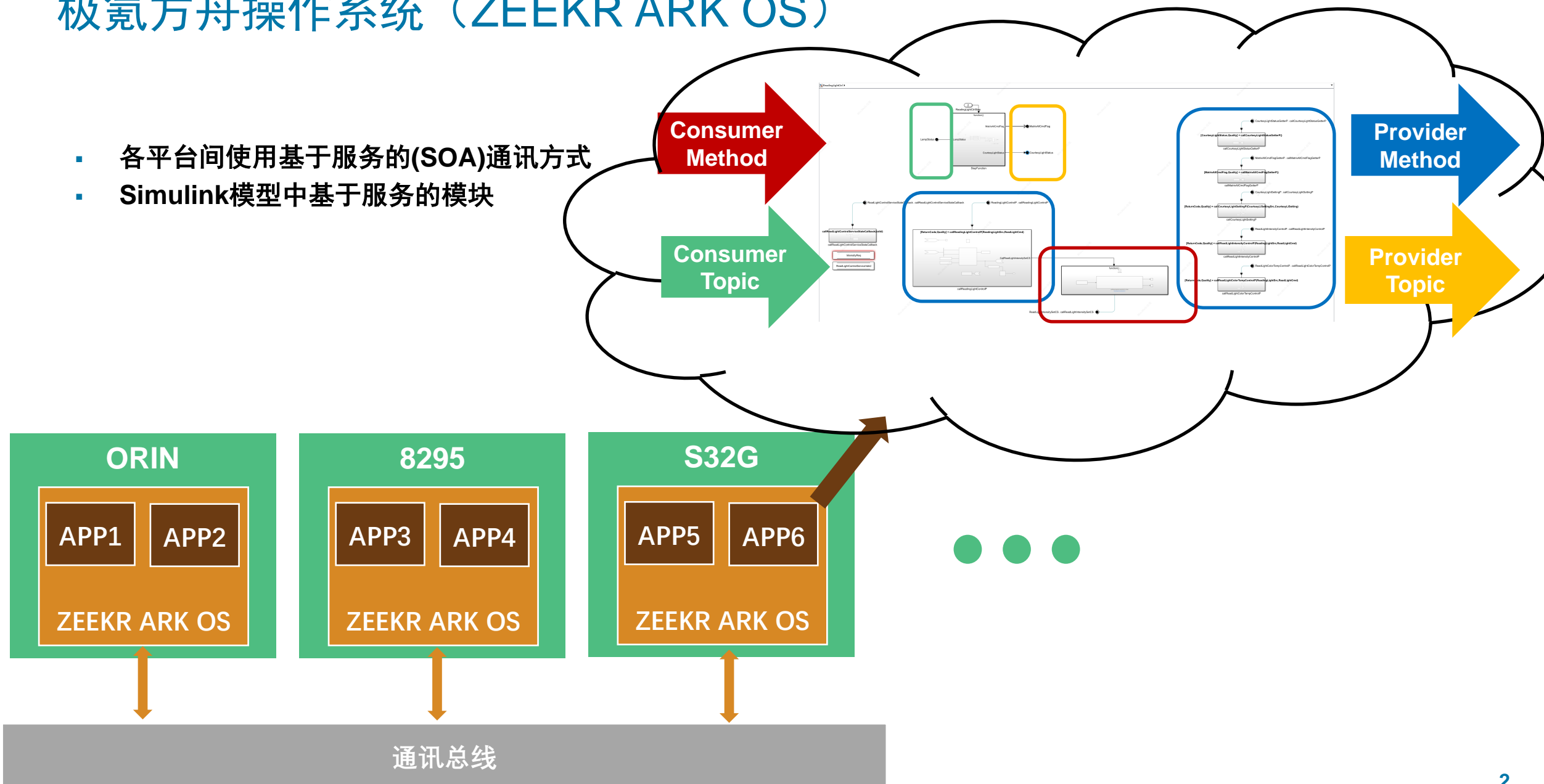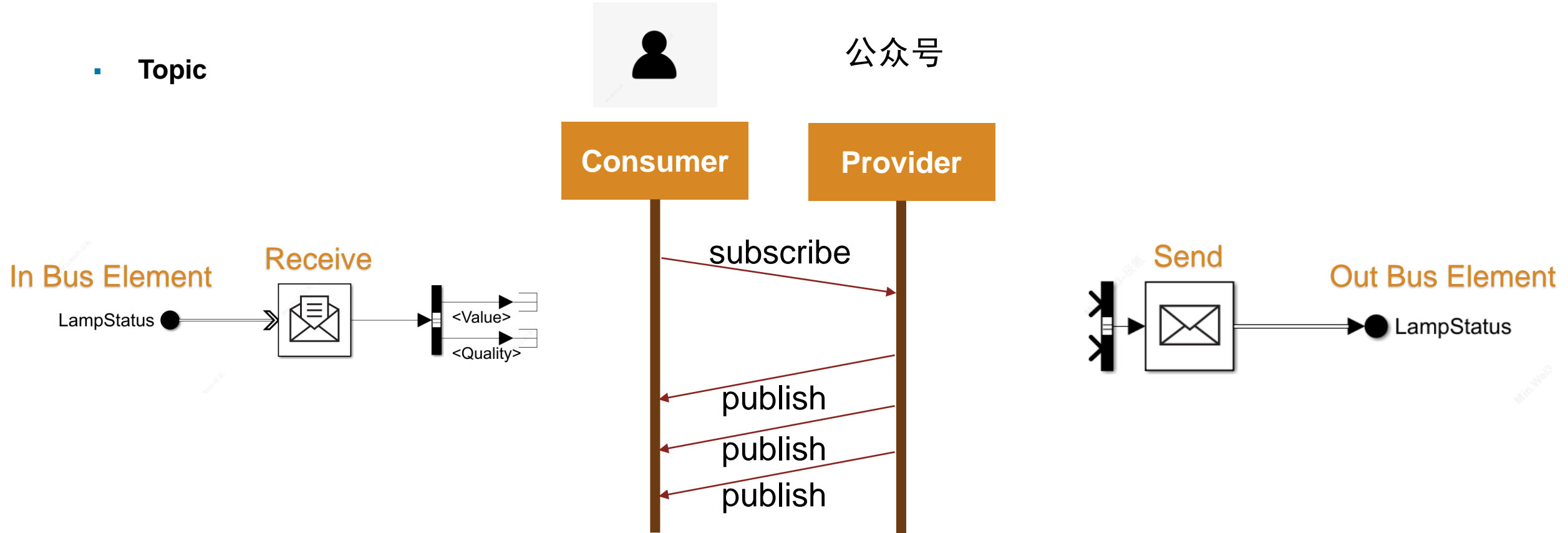*魏旻, 极氪智能科技*

1.背景介绍
2.框架模型配置
3.根据需求建模
4.模型测试
5.代码集成与测试

极氪方舟操作系统（ZEEKR ARK OS）

- 各平台间使用基于服务的(SOA)通讯方式
- **Simulink**模型中基于服务的模块

Simulink中服务化的接口
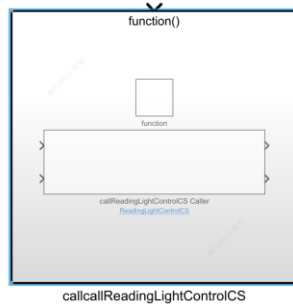
- **Topic**

公众号

Simulink中服务化的接口
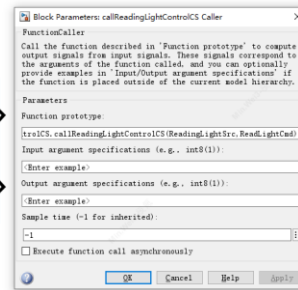
Function Element

ReadingLightControlP . callReadingLightControlP

[ReturnCode,Quality] = callReadingLightControlP(ReadingLightSrc,ReadLightCmd)

callReadingLightControlP

Simulink Function

callReadingLightControlP

- **Method**

function()

function

callReadingLightControlCS Caller
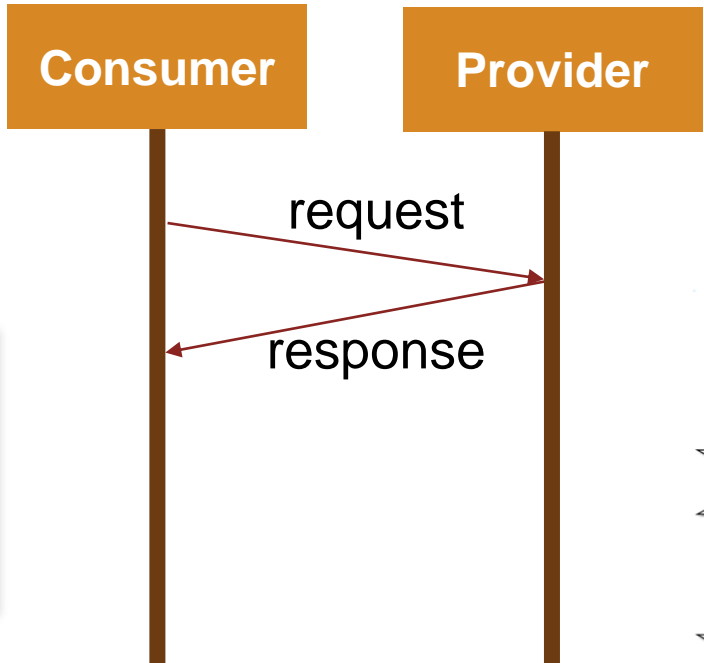ReadingLightControlCS

callcallReadingLightControlCS

ReadingLightControlCS . callReadingLightControlCS

Function Element Call

"一问一答"

| **Consumer** | **Provider** |

request

response

f()

function

ReadingLightSrc   ReturnCode

caller
ReadingLightControlCS.callReadingLightControlCS()

Function Caller

ReadLightCmd   Quality

callReadingLightControlCS Caller

ReadingLightControlCS

Block Parameters: callReadingLightControlCS Caller

FunctionCaller

Call the function described in 'Function prototype' to compute output signals from input signals. These signals correspond to the arguments of the function called, and you can optionally provide examples in 'Input/Output argument specifications' if the function is placed outside of the current model hierarchy.

Parameters

Function prototype:
trolCS.callReadingLightControlCS(ReadingLightSrc,ReadLightCmd)

Input argument specifications (e.g., int8(1)):
<Enter example>

Output argument specifications (e.g., int8(1)):
<Enter example>

Sample time (-1 for inherited):
-1

☐ Execute function call asynchronously

OK   Cancel   Help   Apply

f()

callReadingLightControlP
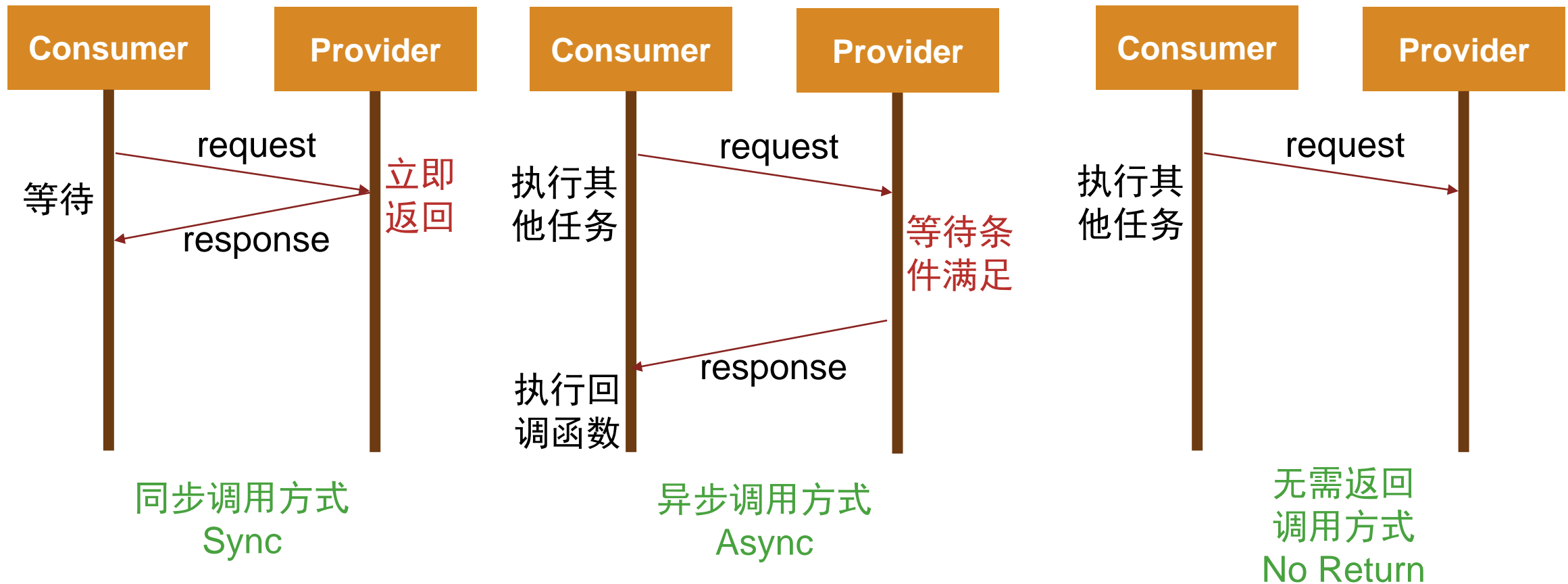
ReadingLightSrc

1

> 0

ReturnCode

0

ReadLightCmd

Quality

Simulink中服务化的接口

- **Consumer Method的**三种调用方式
- **Provider Method的**两种返回方式

同步调用方式
Sync

异步调用方式
Async

无需返回
调用方式
No Return

整体开发流程

系统需求

系统设计

服务定义

需求建模

单元测试

模型测试

代码测试

系统测试

开发过程依赖的组件

系统需求

通讯框架

服务框架

服务框架
生成工具

基础功能

执行管理器	WatchDog	持久化
健康管理器	时间同步	诊断管理
DataExchange	安全引擎	日志
网络管理	XCP	状态管理器

服务配置文件
（arxml）

MBD SOA开发工具
（SOMOC）

软件集成工具包
（ZMINT）

S-Function
&TLC

Simulink
自动建模工具

代码测试工具

服务验证工具
包(SVT)

软件开发工具
包(SDK)

模型测试工具

Test Harness

系统测试

1.背景介绍
2.框架模型配置
3.根据需求建模
4.模型测试
5.代码集成与测试

系统需求

系统测试

系统设计

代码测试

服务定义
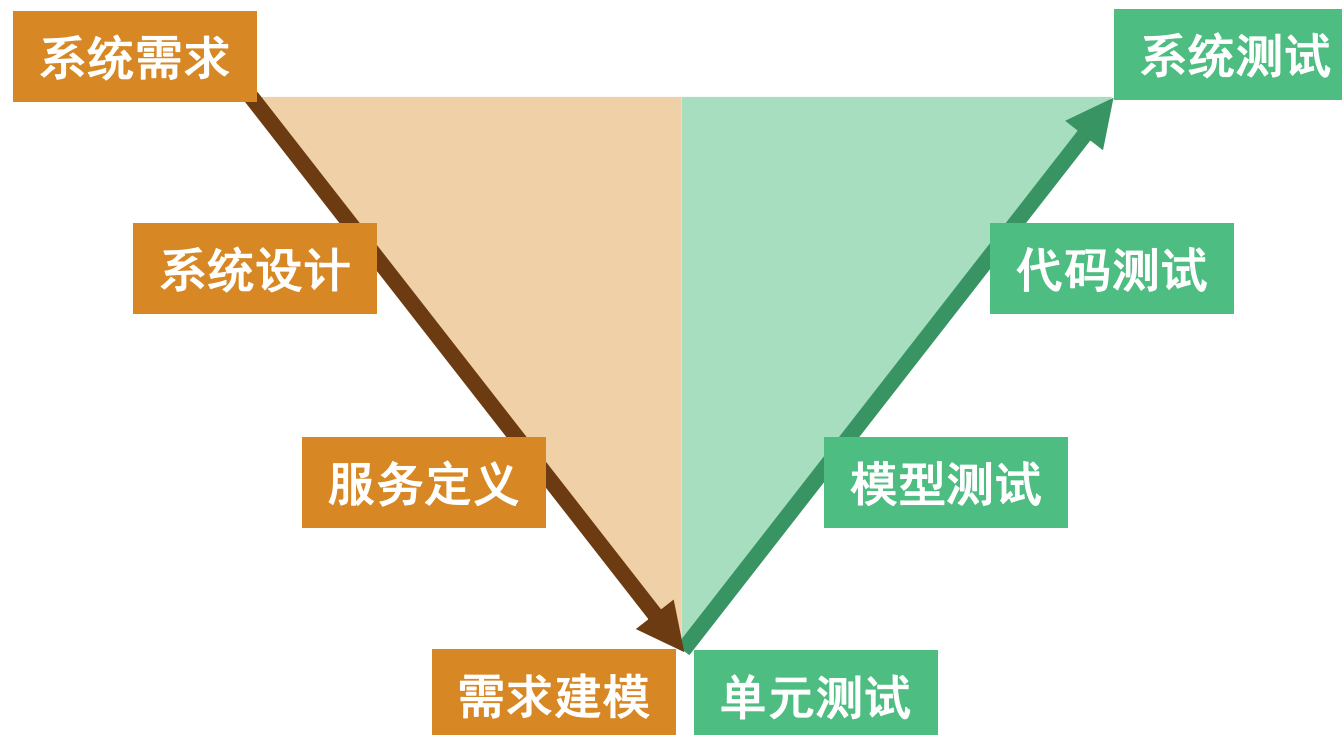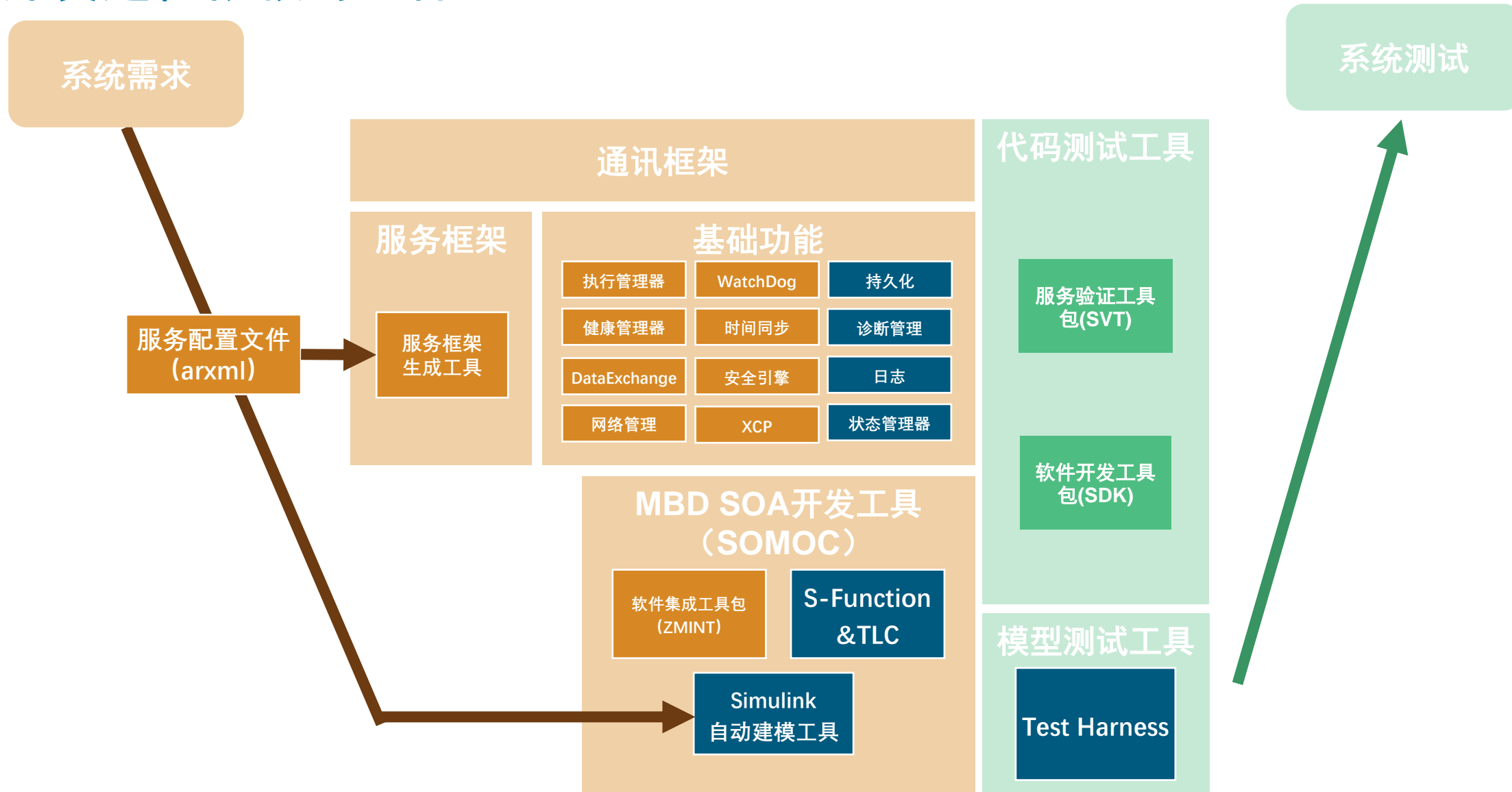
模型测试

需求建模

单元测试
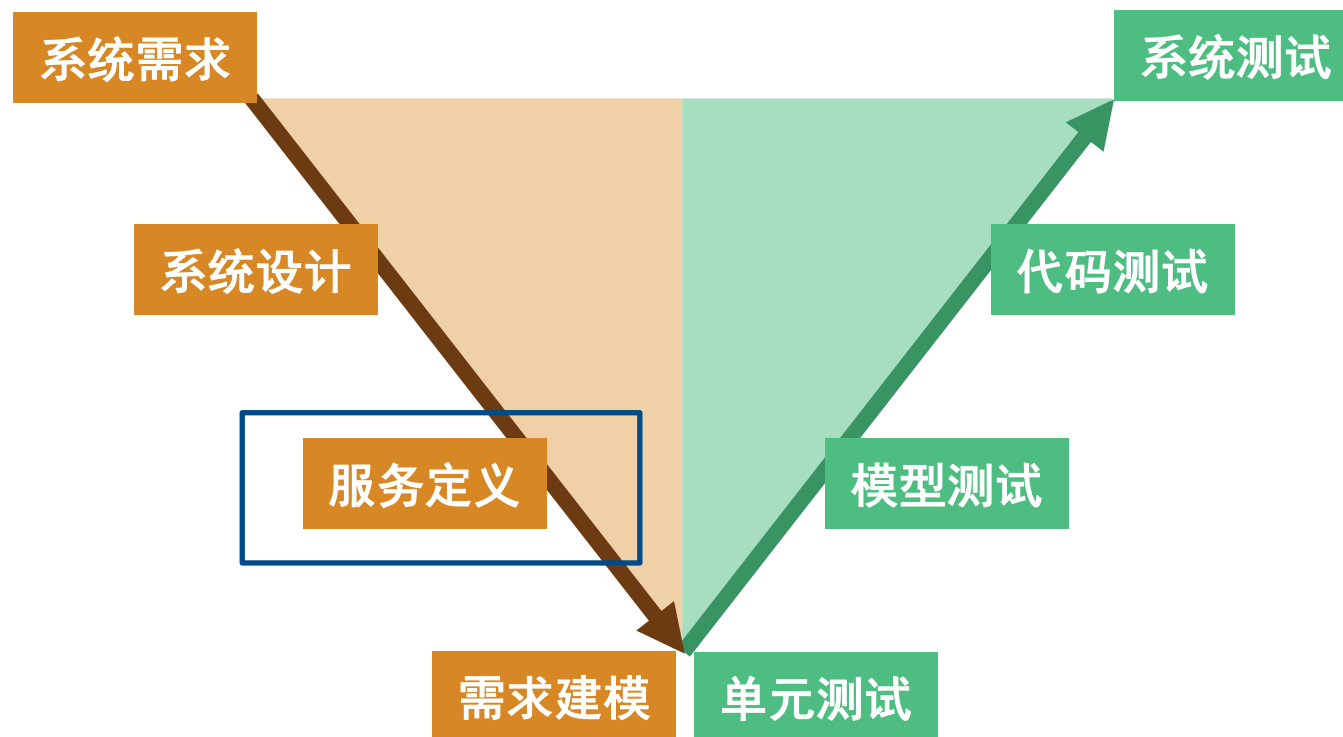
框架模型配置

- 配置服务接口
  - C端选择需要的Method Topic
  - C端Method的执行方式（同步、异步、无需返回）
  - C端是否生成状态回调函数
  - P端Method的返回方式（立即返回、条件返回）
- 配置定时器
  - 间隔时间
  - 起始时间
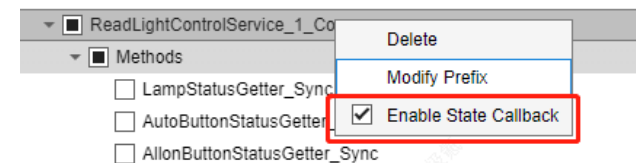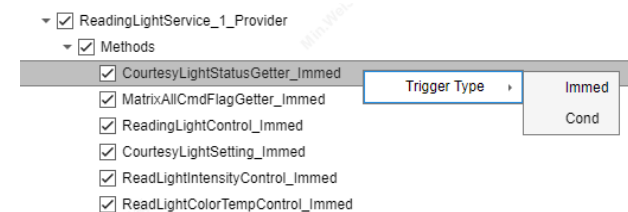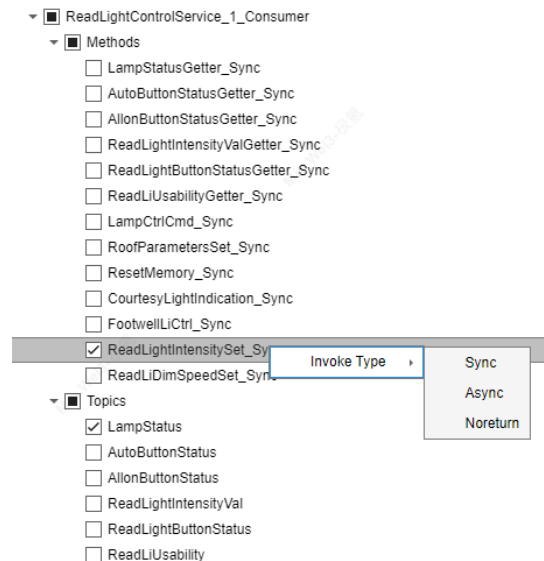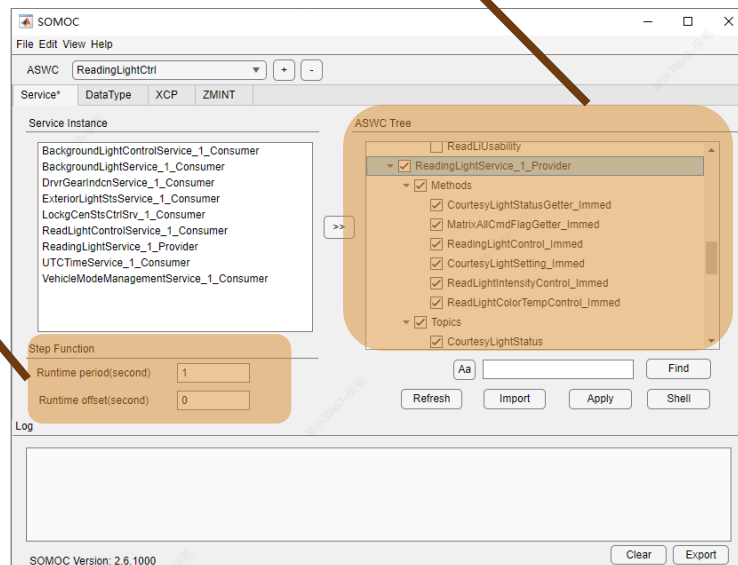
框架模型配置

- 配置特殊数据类型
    - 在Simulink中实现动态数组的最大长度
    - 在Simulink中实现字符串的最大长度
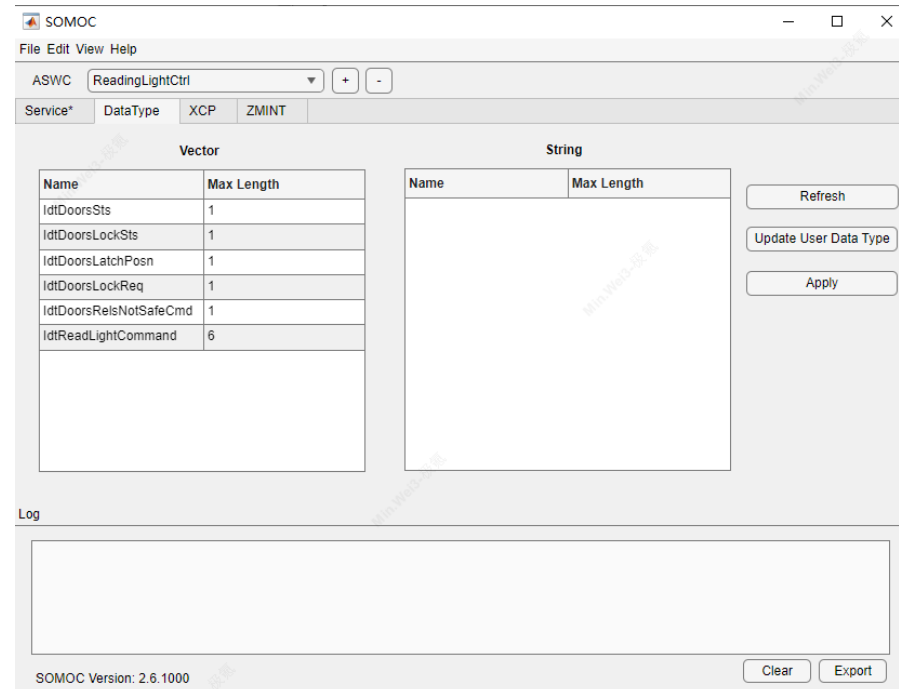
框架模型配置

- 生成框架模型

1.背景介绍
2.框架模型配置
3.根据需求建模
4.模型测试
5.代码集成与测试

系统需求

系统设计

服务定义

需求建模　单元测试

系统测试

代码测试
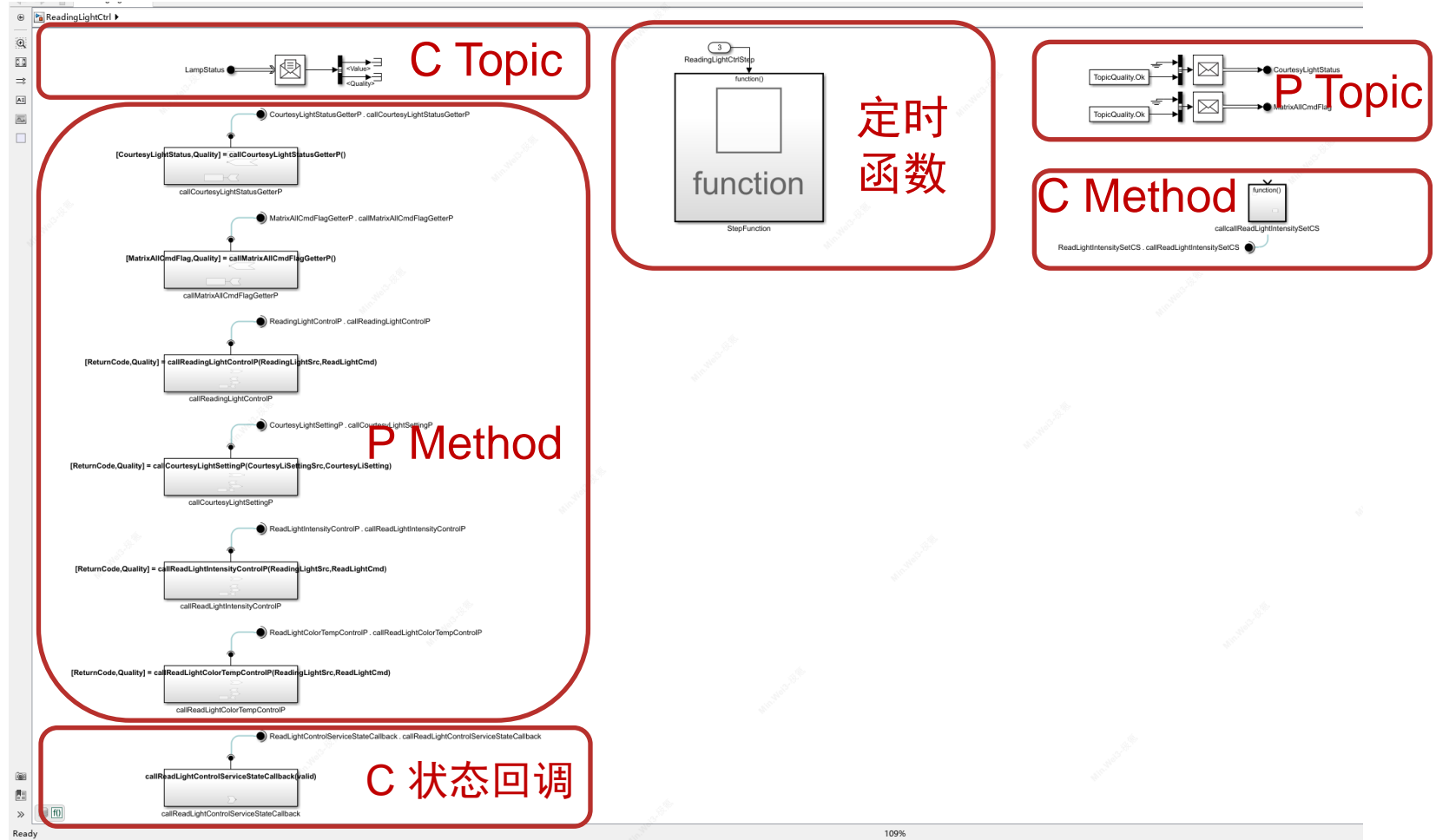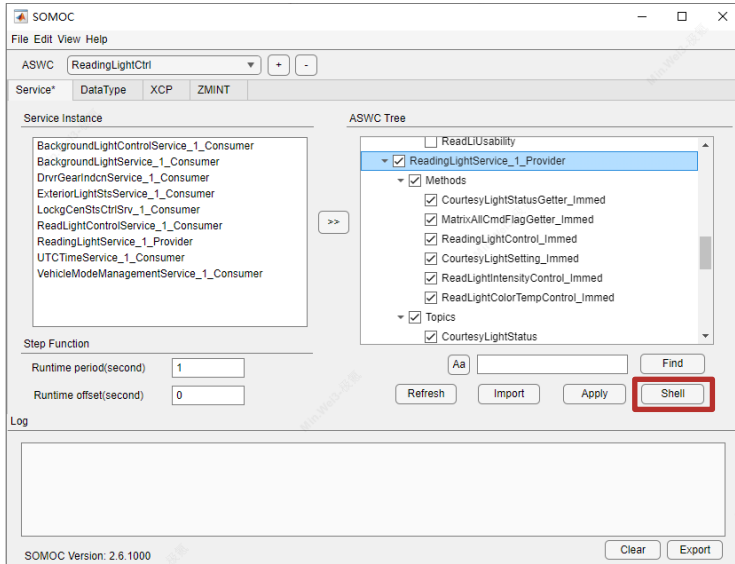
模型测试
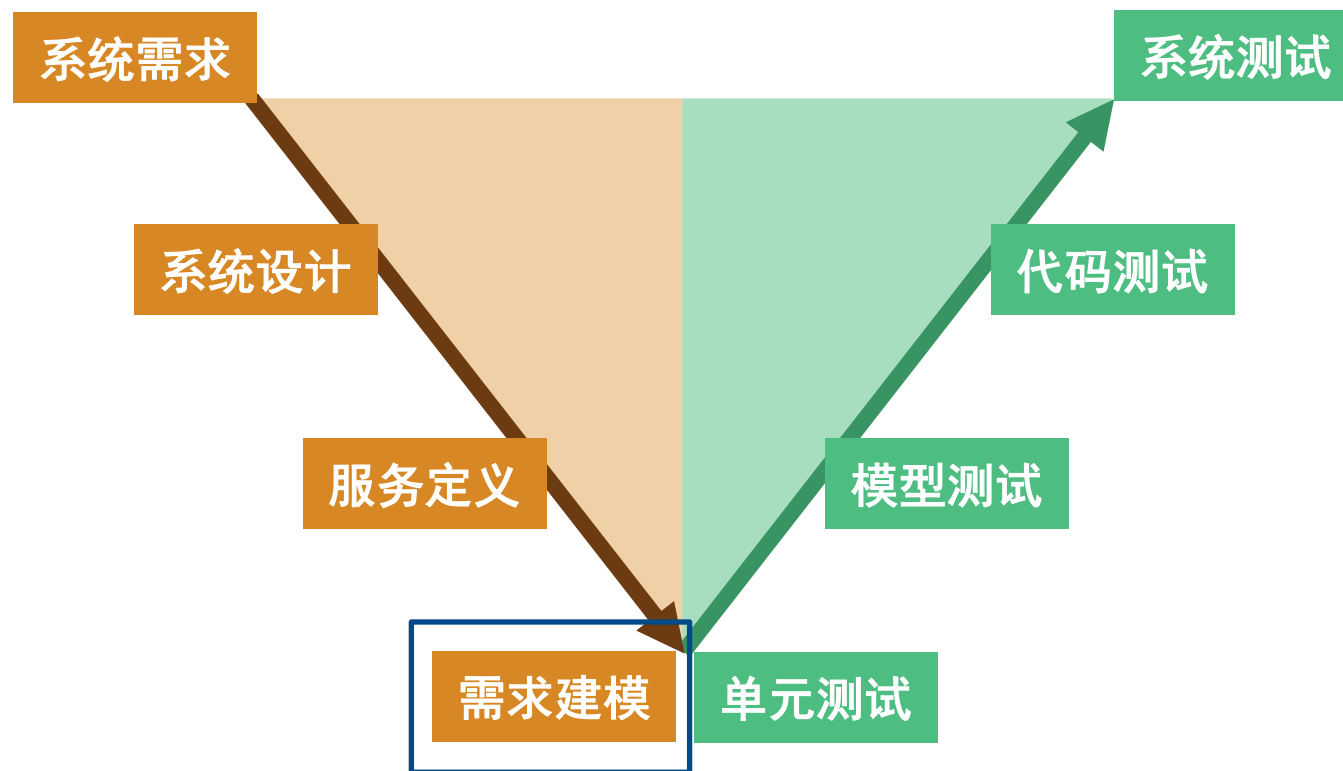
需求建模

- 集成**ZEEKR ARK OS**中间件模块
- 集成定制化业务模块

基础功能

执行管理器	WatchDog	持久化
健康管理器	时间同步	诊断管理
DataExchange	安全引擎	日志
网络管理	XCP	状态管理器

SOMOC

ZMINT	S-Function &TLC
	Simulink

SOMOC

File Edit View Help

☑ Show Log
ZKOSLib ▶
ZKOS sldd
A2L_EXCEL_Demo

Persistency
Log data
DTC
System time
SM Client

BackgroundLightControlService_
BackgroundLightService_1_Con
DrvrGearIndcnService_1_Consu
ExteriorLightStsService_1_Consumer

持久化 — ZKOS Persistency — Persistency

日志 — ZKOS Log data — Log data

空调逻辑跳转 — ZKOS Logic jump for CCM — Logic jump for CCM

ZKOS System time

ZKOS DTC Monitor — DTC Monitor

ZKOS Pref Mgmt — Pref Mgmt

获取系统时间

诊断管理

个性化设置功能

ZKOS SM Clinet — SM Clinet

状态管理器

PROJECT   FILE   LIBRARY   PREPARE   SIMUL
callReadLightControlServiceStateCallback
ReadingLightCtrl ▶ callReadLightControlServiceStateCallback

f()

callReadLightControlServiceStateCallback

valid → ReadLightControlServiceValid

zkos log data
ZKOSLogData

Block Parameters: ZKOSLogData
S-Function (mask)
Parameters
Log type Info
☑ Enable input
Log massage before input "kos log data example, input = "
Log massage after input " zkos log data end."

OK   Cancel   Help   Apply

需求建模

- **S-Function&TLC**

需求建模

- 支持以太网XCP的A2L文件生成（可选）
  - 使用Excel管理标定量与观测量，可以导出A2L文件
  - 在模型中创建标定量与观测量

需求建模

- 根据实际业务需求，搭建**Method**和**Topic**的运行逻辑

1.背景介绍
2.框架模型配置
3.根据需求建模
4.模型测试
5.代码集成与测试

系统需求

系统测试

系统设计

代码测试

服务定义
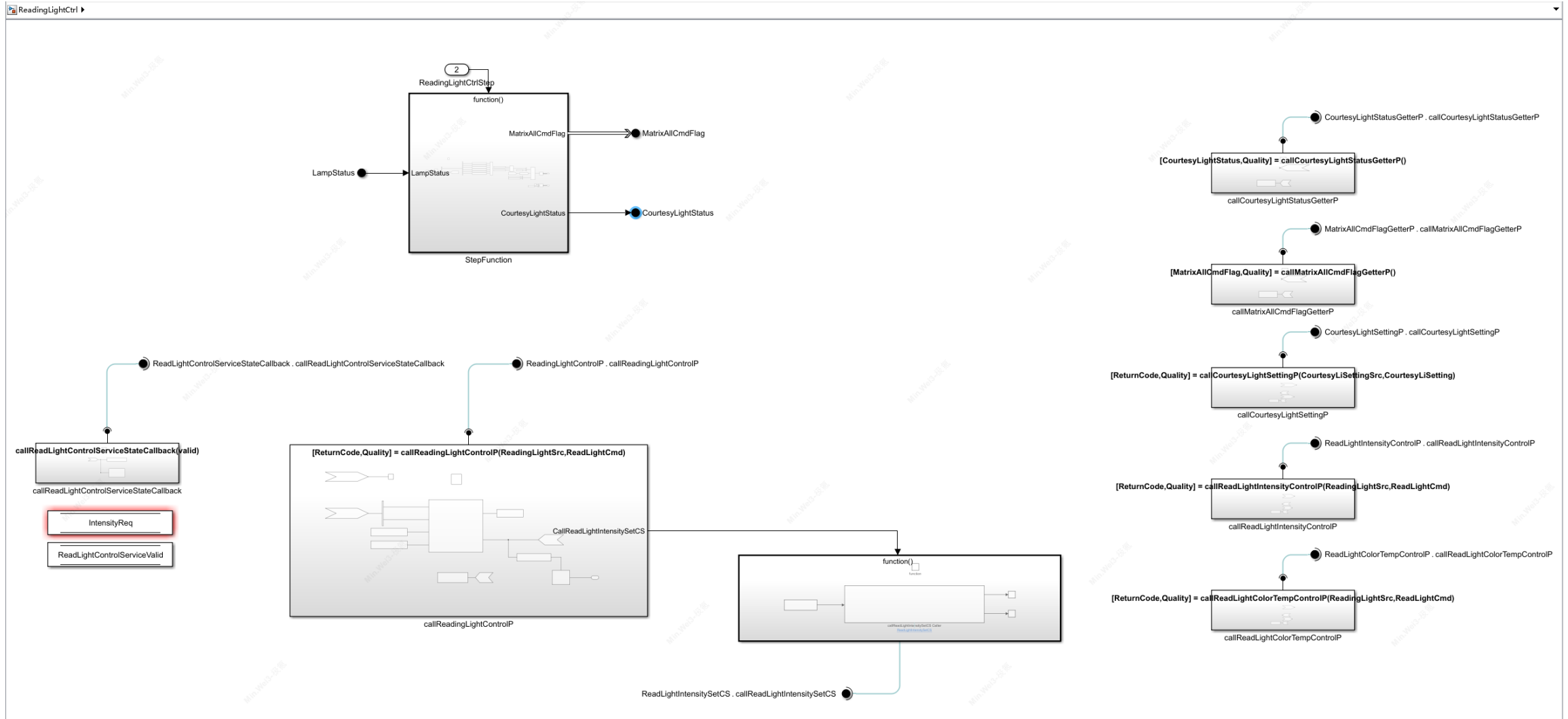
模型测试

需求建模
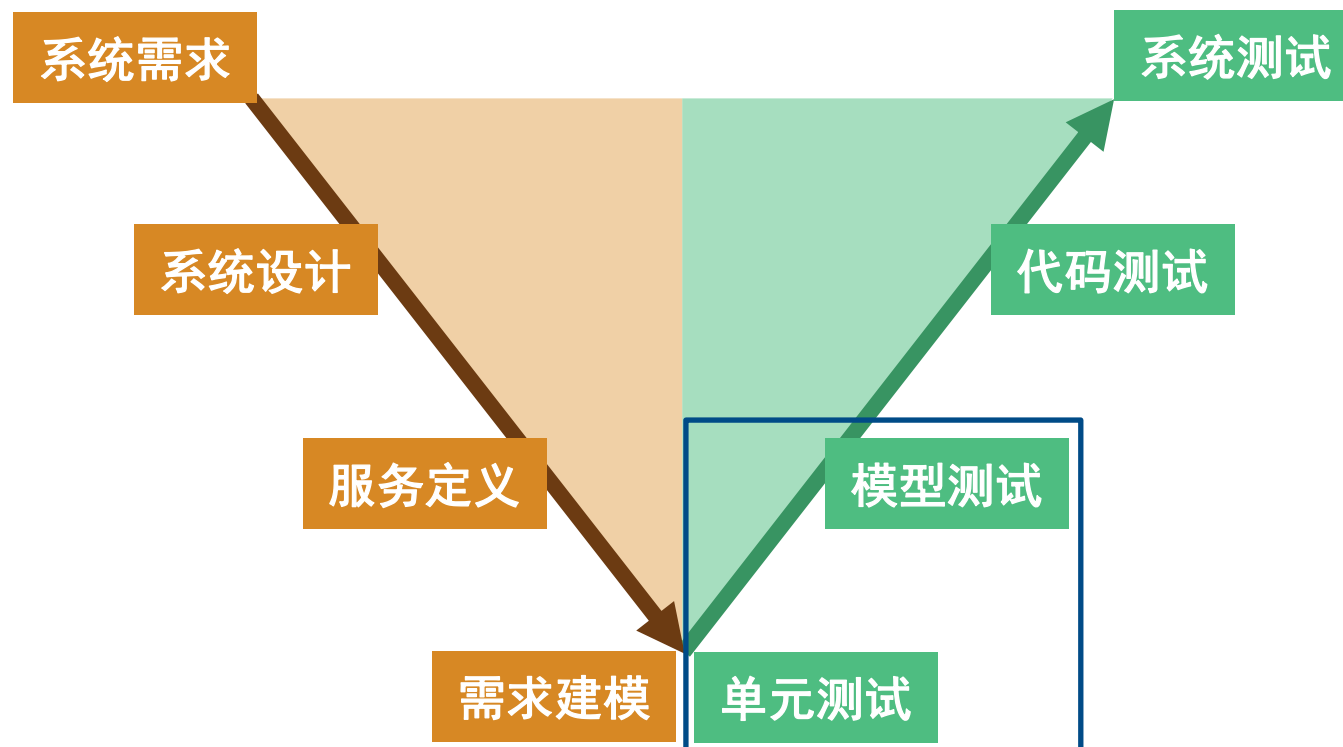
单元测试

单元测试

- **Test Harness**

单元测试

- **Test Harness**

服务模型测试

▪ 建立对端模型

定时器

服务A_C

服务B_P

服务B_C

服务A_P

1.背景介绍
2.框架模型配置
3.根据需求建模
4.模型测试
5.代码集成与测试

生成Simulink代码并集成到ZEEKR ARK OS

- SOMOC自动配置模型参数

Target selection

System target file:	ert.tlc
Description:	Embedded Coder
Shared coder dictionary:	<empty>
Language:	C++
Language standard:	C++11 (ISO)

Build process

☑ Generate code only
☑ Package code and artifacts

Toolchain:	AUTOSAR Adaptive	CMake
Build configuration:	Faster Builds

▸ Toolchain details

Code generation objectives

Prioritized objectives:	Unspecified
Check model before generating code:	Off

Property Inspector

BlockDiagram

Properties | Info | **Execution**

▼ **Type**

Specify execution domain for the selected subsystem or model.

☑ Set execution domain

Domain: Export function
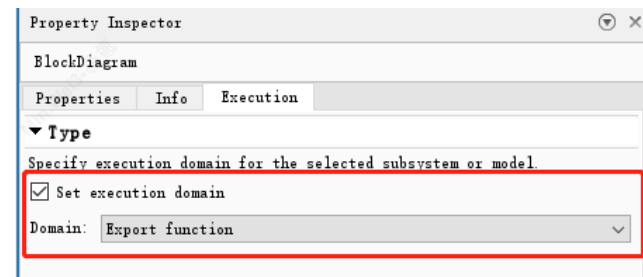
R2024a

导出函数模型概述

导出函数模型是 Simulink® 模型，它们可以生成可与外部环境和调度器集成的独立函数的代码。函数是使用 Function-Call Subsystem、函数调用 Model、Simulink Function、Message Triggered Subsystem 和 S-Function 模块定义的。

以下导出函数模型包含使用 Function-Call Subsystem 模块定义的两个函数。有关创建此模型的分步过程，请参阅Create Export-Function Model。
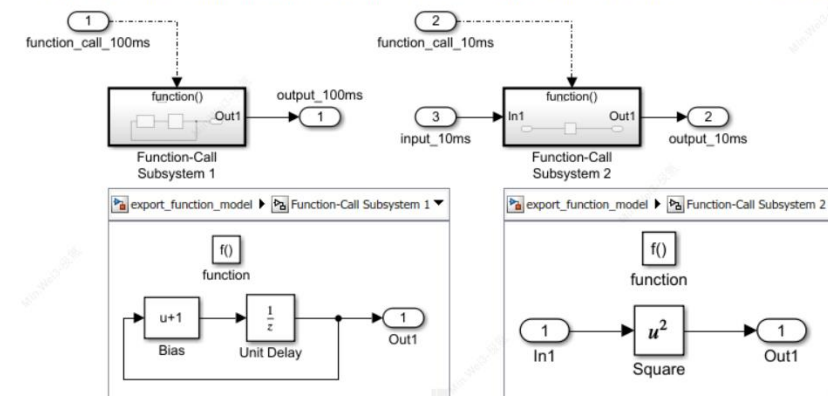
生成Simulink代码并集成到ZEEKR ARK OS
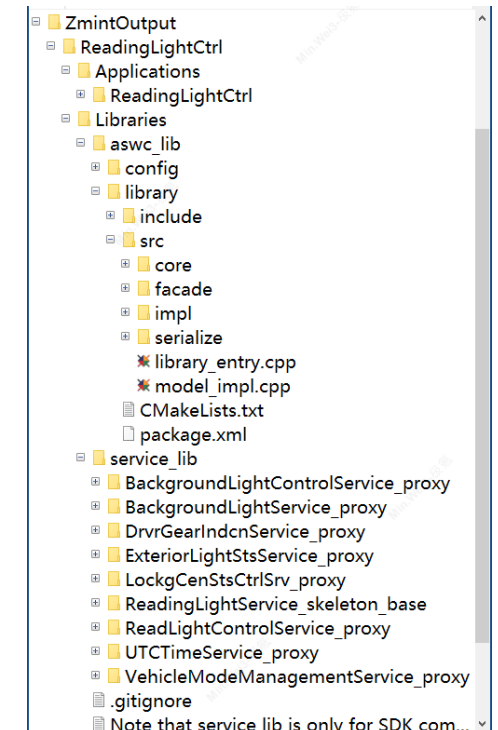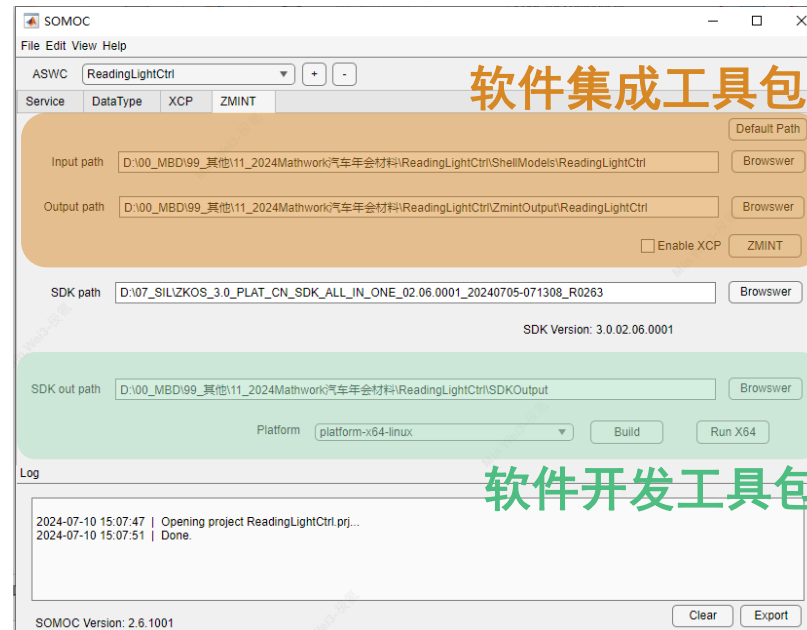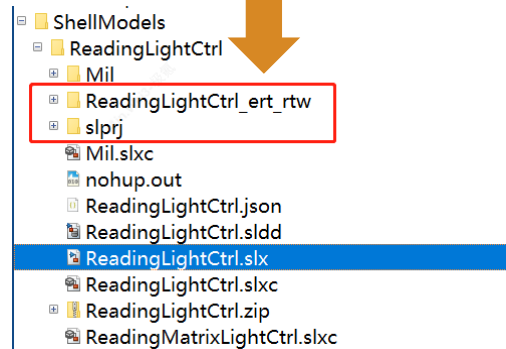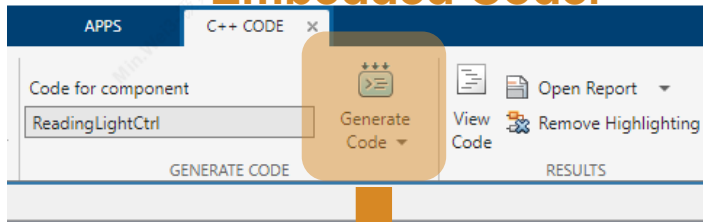
代码测试

- 编译与运行APP

软件编译结果

软件运行结果

支持跨平台编译

Windows/Ubuntu

容器
（ZEEKR ARK OS）

APP

服务验证工具包
（SVT）

代码测试

- 服务验证工具包（SVT）

代码测试

- **SVT 代码测试**

代码测试

- **SVT 代码测试**

6.总结

- ZEEKR ARK OS与MATLAB/Simulink结合
- 根据服务信息自动生成服务化的框架模型
- 使用S-Function&TLC封装ZEEKR ARK OS中间件模块
- 使用Test Harness进行测试
- 使用ZEEKR ARK OS的软件开发与测试工具包进行代码测试

系统需求

系统测试

系统设计

代码测试

服务定义

模型测试

需求建模　单元测试
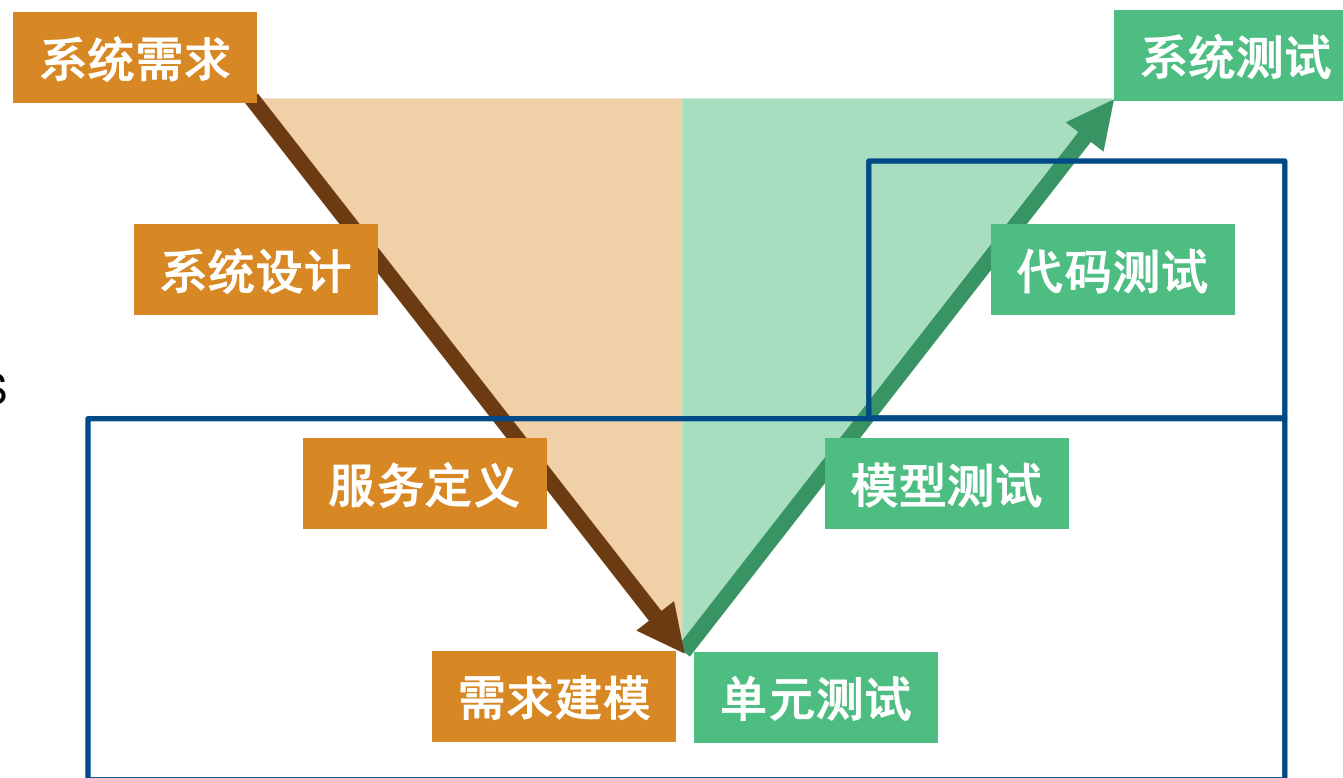
2024 MathWorks
中国汽车年会

Thank you

MathWorks®