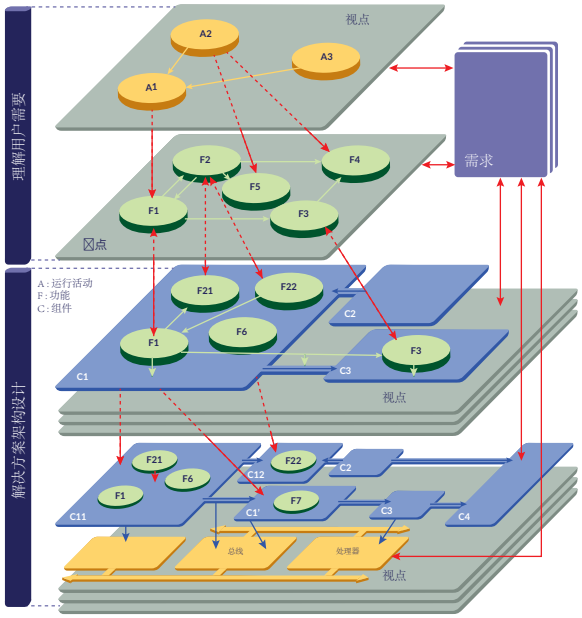
ARCADIA

mbse-capella.org

一个定义、分析、设计和验证系统软件及硬件架构的工具方法

支持工程领域的高效协作

根据运行需要对解决方案进行确认或修正，简化影响性分析过程



运行分析
系统的用户需要完成哪些任务

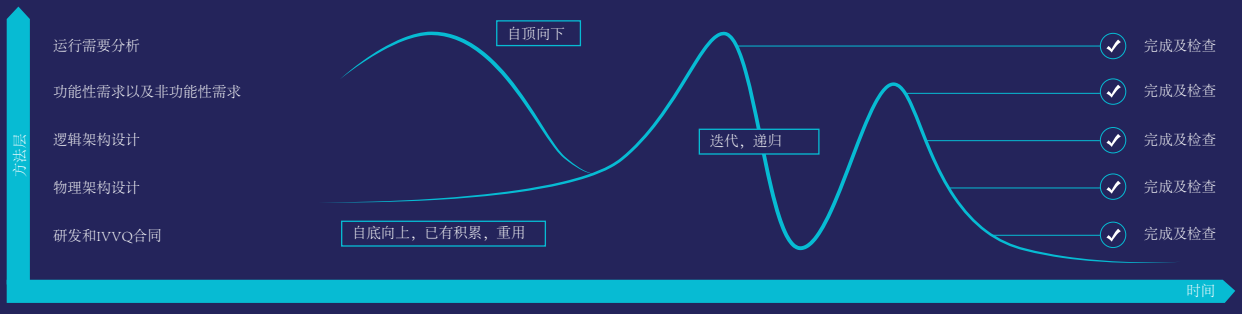
功能性需求和非功能性需求
系统要为用户完成什么

逻辑架构
该系统将如何实现预期的功能

物理架构
该系统将如何研制与搭建

与大多数流程兼容

自顶向下、自底向上、迭代、改进型、混合……



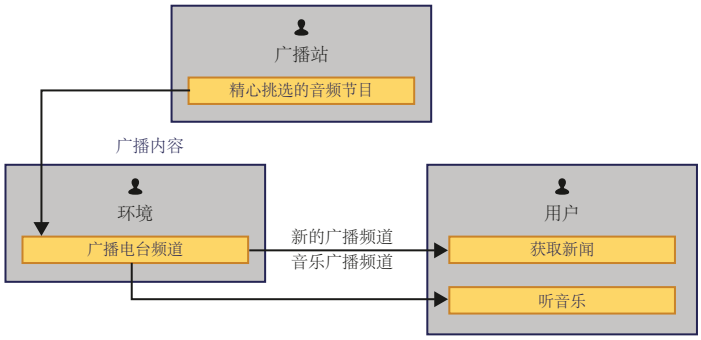
方法步骤 | 任务 | 示例模型

用户需要

用户运行需求分析

系统的用户需要完成哪些任务

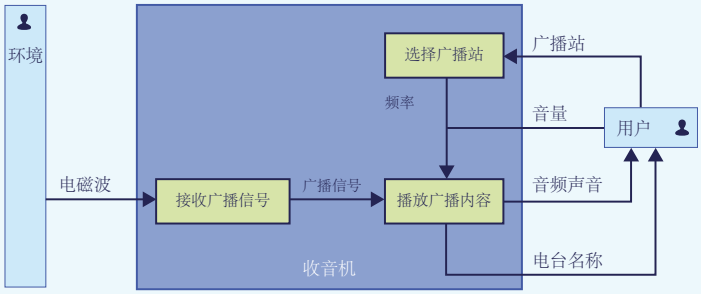
- ✓定义运行能力
- ✓开展运行需求分析



系统/软件/硬件需求分析

系统要为用户完成什么

- ✓对能力开展权衡分析
- ✓开展功能性和非功能性分析
- ✓梳理并整合需求

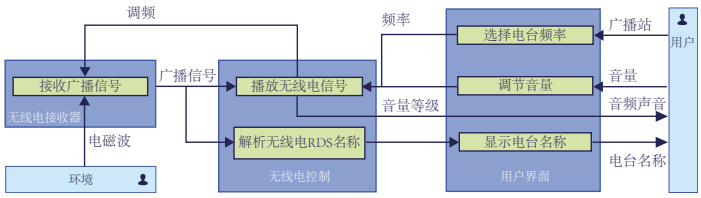


解决方案

逻辑架构设计

该系统将如何实现预期的功能

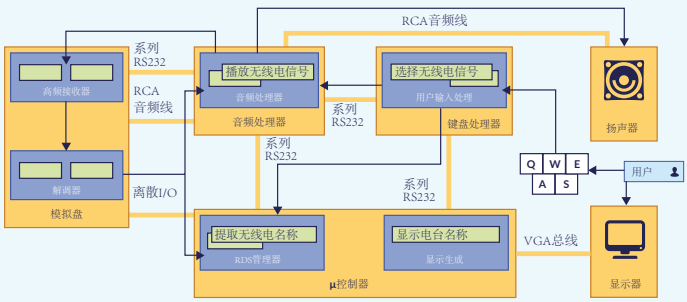
- ✓定义架构驱动和视点
- ✓构建备选架构的组件分解方案
- ✓选择最佳的折中方案



物理架构设计

该系统将如何研制与搭建

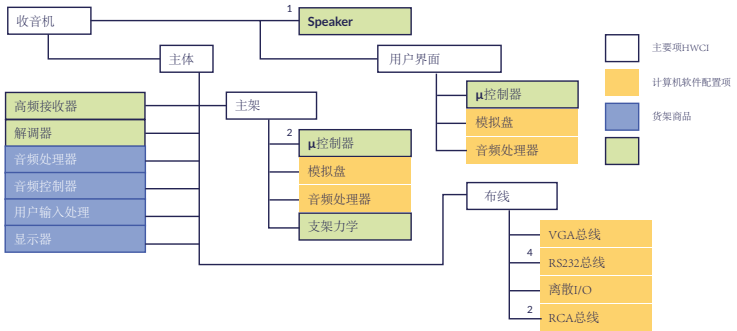
- ✓定义架构样式
- ✓评估重用已有的资产
- ✓定义一个物理参考架构
- ✓验证并检查它



研发合同

对每个设计师/分包合同的期望是什么

- ✓定义一个组件的IVVQ策略
- ✓定义并执行一个PBS和组件集成合同



概念

- 运行能力
- 施动者、运行实体
- 施动者活动
- 运行活动和施动者之间的交互
- 活动及交互中使用的信息
- 运行流程相关活动
- 动态行为场景

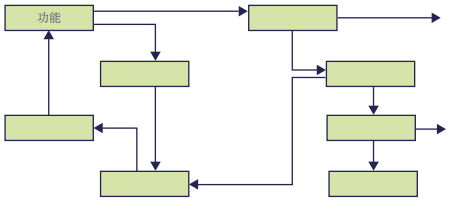
- 施动者，系统，能力
- 系统功能和施动者功能
- 功能之间的数据流交换
- 数据流中的功能链路
- 功能、功能交互、数据模型中使用的信息
- 动态行为的场景
- 模式和状态

相同的概念，以及：
- 组件
- 组件端口及接口
- 组件之间的交互
- 组件上分配的功能
- 通过分配的功能交换确定组件的接口

相同的概念，以及：
- 行为组件细化了逻辑组件，
并实现了功能行为
- 实现组件为行为组件提供资源
- 物理组件之间的物理链接

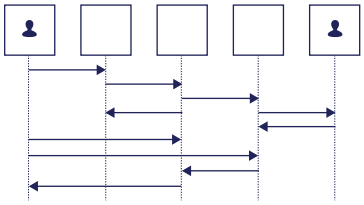
- 配置项树
- 零件编号、数量
- 研发合同(预期行为、接口、场景、资源消耗、非功能属性……)

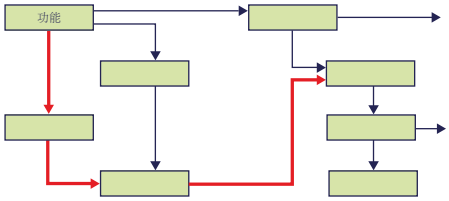
描述方式



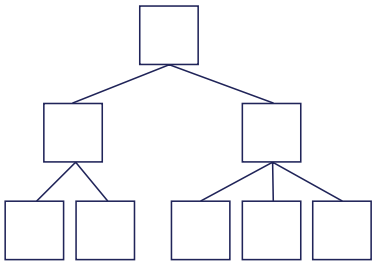
数据流：
功能之间的交换和运行活动之间的交互

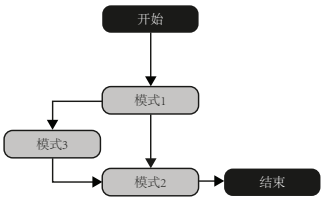
场景：
施动者、系统、组件之间的交互和交换





功能链路，运行流程：
通过功能或运行活动实现

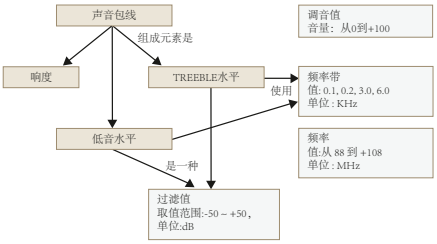


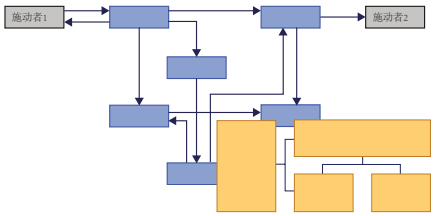


模式和状态：
从属于施动者、系统、组件

功能&组件分解

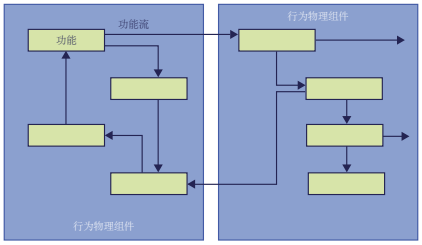
数据模型：
数据流和场景中的内容，
接口的定义和校验



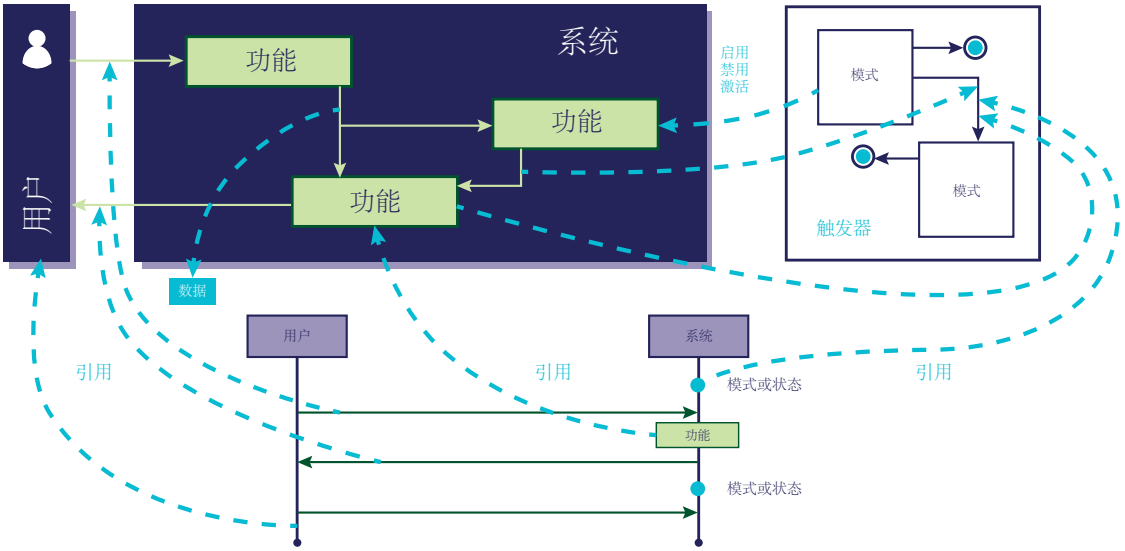


组件布线
所有类型的组件

分配：
运行活动分配到施动者、功能分配到组件、行为逻辑组件分配到物理组件、数据流分配到接口、元素分配到配置项中



态行为

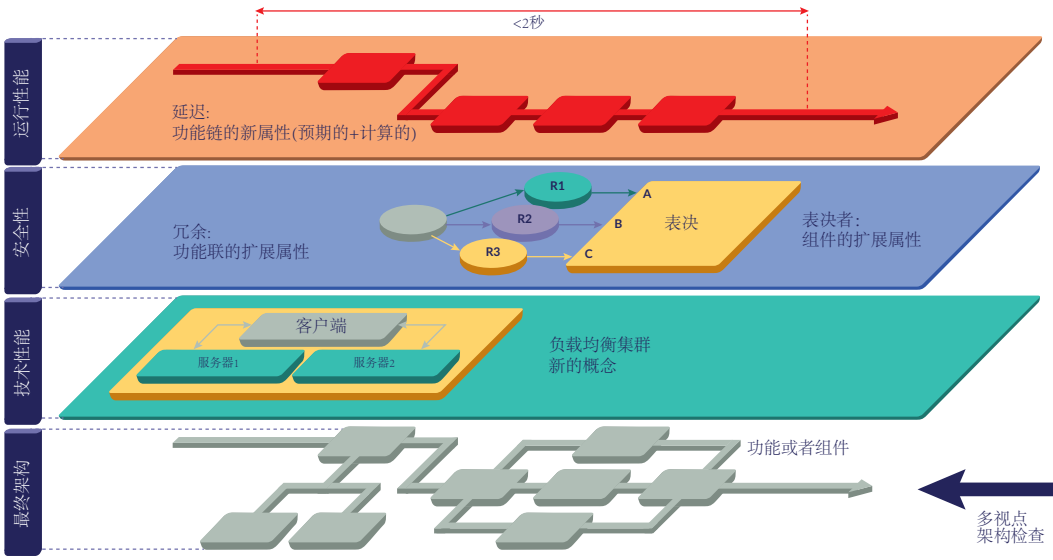


通过邀请行业利益攸关方和原始非功能需求来验证和检查解决方法

方法层级	特定性能数据样本	安全性示例数据
运行需要分析	对威胁的最大反应时间	危害事件
功能/非功能需求分析	威胁反应的功能链(Fc) FC允许的最大时延	与危害事件相关的关键功能链
逻辑架构设计	处理并传递复杂性 功能链路分配	保护功能链的冗余路径
物理架构设计	Fc上的资源消耗 产生的计算延迟	共模失效 Fc的失效传播
研发和IVVQ合同	分配资源以满足延迟	所需的可靠性水平

非功能需求视点（示例）

- ✓ 成本和工期
- ✓ 接口
- ✓ 性能
- ✓ 可维护性
- ✓ 安全性/安保
- ✓ 等等
- ✓ IVVQ
- ✓ 产品政策



架构检查

- 根据物理组件和通信性能计算功能遍历时间
- 推断最终实现的延迟并与预期进行比较
- 根据功能链路的危害等级判断所需的冗余级别
- 将硬件失效传递到功能链路
- 基于功能计算组件负载并检查其充分性
- 包括计算能力和通信带宽

THALES