



产品介绍

北京马到科技有限公司

北京马到科技有限公司

二〇二一年四月

目录

1. MateBus(马到总线).....	4
1.1. 概述.....	4
1.2. 系统特征.....	4
1.3. 系统架构.....	5
1.3.1. 通信部分.....	5
1.3.2. 框架部分.....	6
1.4. 系统功能.....	8
1.5. 建成后的成果.....	11
1.5.1. 实现软、硬件互连互通.....	11
1.5.2. 提供构件化软件系统框架.....	11
1.5.3. 提供分布式应用系统引擎.....	11
1.5.4. 提供运维管控支持.....	12
2. Adapters(设备接口适配器).....	12
2.1. 概述.....	12
2.2. 设备接口集成规范.....	13
2.3. 多协议设计工具.....	14
2.4. 代码生成器.....	14
2.5. 设备集成设计工具.....	15
2.6. 建成后的成果.....	16
3. MateDDS(数据分发服务).....	16
3.1. 主要特点.....	16
3.2. 同类产品比较.....	16
3.3. MateDDS 扩展功能.....	17
3.4. 建成后的成果.....	18
4. Manager(管理器).....	18
4.1. 远程探测.....	19
4.2. 远程监视.....	19
4.3. 远程控制.....	20
4.4. 远程配置.....	20
4.5. 远程部署.....	21
4.6. 远程调试.....	21
4.7. 运行管理.....	21
4.7.1. 仿真导调.....	21
4.7.2. 测试导调.....	22
4.7.3. 训练导调.....	22
4.8. 数据记录.....	22
4.9. 数据回放.....	22
4.10. 资源调度与管理.....	22
5. Tools(工具).....	23
5.1. 通用工具.....	23
5.1.1. 协议设计工具.....	23
5.1.2. 业务设计工具.....	24

5.1.3.	部署设计工具.....	28
5.2.	专用工具.....	35
6.	Cases(案例).....	35
6.1.	XXX 作战原型实验室建设.....	35
6.1.1.	系统概述.....	35
6.1.2.	系统组成.....	36
6.1.3.	系统实现.....	36
6.1.4.	Mate 在系统中的应用	37
6.2.	XXX 作战协同实验室建设.....	37
6.2.1.	系统组成.....	38
6.2.2.	Mate 总线在系统中的应用	38
6.3.	XXX 水下仿真系统	38
6.3.1.	系统概述.....	38
6.3.2.	系统组成.....	39
6.3.3.	系统实现.....	39
6.3.4.	Mate 总线项目中的应用	40
6.4.	XXX 构件化战场仿真系统.....	40
6.4.1.	系统架构.....	40
6.4.2.	系统部署.....	41
6.4.3.	系统构件框图.....	42
6.5.	XXX 舰载机起降模拟训练系统.....	43
6.5.1.	系统组成.....	44
6.5.2.	系统部署.....	44
6.5.3.	系统实现.....	46
6.5.4.	Mate 总线在系统中的应用	47
6.6.	XXX 通信数据链模拟训练系统.....	47
6.6.1.	系统架构.....	47
6.6.2.	系统布局.....	48
6.6.3.	Mate 总线在系统中的应用	49
6.7.	自动化测试系统.....	49
6.7.1.	测试系统功能.....	49
6.7.2.	系统组成.....	49
6.7.3.	系统应用.....	50
6.7.4.	Mate 总线在系统中的应用	51

1. MateBus(马到总线)

1.1. 概述

MateBus 一套分布式集成运行环境，它集成了多种通信中间件和通信协议，解决了不同设备之间的差异、操作系统之间的差异、通信协议和通信方式之间的差异、软件体系结构之间的差异。

使用统一通信总线，统一的通信接口和集成工具，实现多种通信中间件、通信协议的在系统中免编程的无缝集成。

提供各种物理设备接入规范，为实现物理设备免编程的无缝接入提供接口。

提供了计算任务（数字仿真）接口规范，支持分布式数字仿真。提供了显控集成接口规范，支持各种分布式显控台接入。

提供各种分布系统的引擎框架，如数字仿真引擎、测试引擎、模拟训练引擎等。

提供异构系统接入规范，为支持总线型结构、星型结构，发布-订阅模式、请求应答模式、消息队列等不同结构的软件及软件系统的集成提供接口。

实现各种设备、各种软件的数据采集，为作战系统、电力系统、测试系统、效能评估系统、以及后期的大数据和人工智能提供数据支持。

1.2. 系统特征

- 1) **实时性高：**系统集成各种高实时性的通信中间件（如 DDS），提供开放式接口，可以集成多种实时总线和通信接口，如 1553B、CAN、反射内存卡等
- 2) **集成范围广：**该分布式集成框架可以集成多种通信中间件，支持多种通信协议，能够将各种各样的分布式应用程序在异构的环境中进行集成。
- 3) **耦合性低：**本集成框架使用通信适配器的方式，将各种通信中间件、实时总线和通信接口集成在一起，屏蔽了硬件、操作系统、通信协议、通信方式之间的差异，使用发布订阅的匿名通信，各应用程序不再与通信结合紧密，应用程序之间耦合松散。
- 4) **使用点到点的总线结构：**各通信节点之间的为对等关系，没有中心概念。
- 5) **使用匿名通信：**各节点之间的通信为匿名通信，与通信协议无关、与通信方式无关，只与感兴趣的数据有关，不关心数据的来源，也关心数据去处，更不关心数据何时发送、

何时接收。

- 6) **集成化用户界面:** 分布式应用程序业务功能使用组件技术来实现, 使用统一的集成化用户界面, 统一的运行方式、统一的风格。
- 7) **统一的建模工具:** 系统中通信接口、应用程序各业务组件、分布式系统的设计、应用程序的配置等由统一的建模工具来完成, 生成相应环境下的代码、封装了各通信中间件的使用、组件技术的使用, 大大降低了系统设计、开发、调试和集成的难度, 减少了设计、开发的工作量。
- 8) **提供集中管理的工具:** 提供分布式系统的远程监视、远程控制、远程部署、远程调试的工具, 便于分布式系统的测试、调试、集成和运行管理。

1.3. 系统架构

MateBus 是位于应用层之下、操作系统之上的一层软件, 如下图所示, 用于解决各种分布式系统的之间的互联互通, 同时为各种分布式系统提供应用框架。它由两大部分组成, 一部分为通信部分, 另一部分为框架部分。



1.3.1. 通信部分

通信部分定义了各通信中间件或者通信协议的集成规范, 可以将第三方通信中间件无缝的、免编程的集成到 MateBus 中, 实现应用程序使用多种通信中间件或者通信协议与其它的应用程序进行通信。

通信部分可以集成 DDS、HAL、ZMQ、MQTT 等通信中间件，也可以集成 UDP、TCP 通信协议，还可以是 ORB 或者数据库。

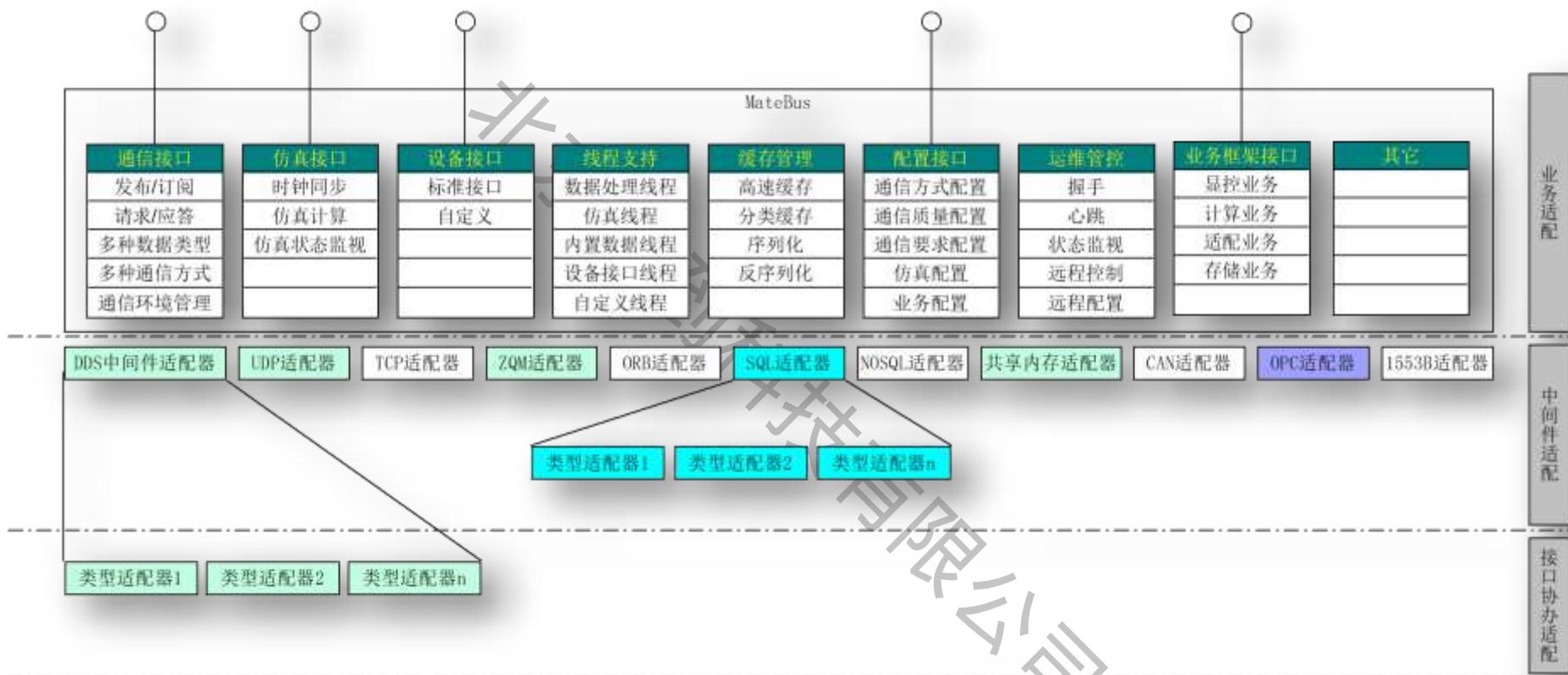
系统提供多种通信中间件的适配器和这些通信中间件的代码生成器，可以直接将 IDL 文件转换成可以成动态库，供 MateBus 使用。

1.3.2. 框架部分

框架部分定义了基于实时分布式应用程序运行的规范和基本框架。它根据应用程序的类型（显示控制应用程序、数字仿真应用程序、设备适配应用程序、计算任务应用程序），定义并实现了这些应用程序在 MateBus 上运行接口、业务处理流程、动态配置规范等，使得这些应用程序在 MateBus 上即插即用。

MateBus 内部结构如下图所示，它由三层组成。上层为实现分布式应用程序适配层，为应用程序提供接口、调用、通信等方面的规范，各应用程序可以在 MateBus 上集成。中层为通信中间件适配层，可以集成多种通信中间件或者通信协议，应用程序可以使用这些通信中间件或者通信协议进行通信。下层为协议适配层，用于支持不同的数据报文集成、扩充、扩展和升级。

其中上层是固定的，用户可以直接将 MateBus 总线集成到应用程序中。中层为半固定的，它可以根据不同的通信中间件进行扩充，但是必须是手工编写代码，一次编码，永久使用。下层是活动的，不同的应用程序，它们可能不同，它是由 IDL 文件通过本系统提供的代码生成器直接生成。



MateBus 内部架构图

1.4. 系统功能

- ◇ 分布式集成框架提供分布式程序接口和分布式系统运行框架支撑环境。
- ◇ 应用程序接口提供了与平台无关、协议无关、与通信方式无关的数据发送与接收。
- ◇ 在框架支持环境中，提供多个（由应用程序数量决定）应用程序“插座”，应用程序直接插入“插座”中，便可进行数据的发送与接收。
- ◇ 应用程序使用何种通信，由通信适配器的调度器来决定，调度器根据系统要求（如配置信息），决定使用何种通信适配器。
- ◇ 通信配置器以插件方式调用，需要时加载，不需要时不会加载。通信适配器是通信接口的一个实现，一个通信适配器专用于某些通信中间件，如 DDS 通信适配器专用于 DDS 的通信。通过添加或者更新不同的通信适配器，系统可以支持多种通信方式。
- ◇ 系统中的通信方式、通信协议等都是由通信适配器来完成。
- ◇ 当有新通信技术要使用时，只需要开发一套该通信技术的通信适配器，在应用程序的配置文件中指定它即可，系统原有的结构和原来的应用程序不做任何更改。
- ◇ 当要使用老技术时，在应用程序的配置文件中指明使用老技术的通信方式。
- ◇ 当系统中同时使用多种通信中间件进行通信，并且它们之间是互联互通可，可以通过协议转换的方式进行异构系统的集成。

-
- ◇ 系统中应用程序的所有功能都是由组件来完成，它可以是一个组件也可以是多个组件，根据系统的功能以及复杂程序来定；
 - ◇ 每个组件是一个独立的对象，它具有完成功能，与应用程序、操作系统、通信中间件是解耦合；
 - ◇ 各组件存在于组件库中，而组件库是以动态库（在 Linux、VxWorks 中叫共享库）的方式提供，通过配置文件由通用的应用程序动态创建、动态运行；
 - ◇ 组件提供了在该平台上运行的基本原型，如数据的发布-订阅、握手机制、心跳、状态报告、日志信息、远程控制、热备等，也就是新组件如果是在原型组件基础上派生的，新组件具备上述的功能。
 - ◇ 根据应用程序用途，可以将组件分成五大类，即适配业务构件、数据显示组件、数据存储组件、数字仿真组件和数据分析组成，系统提供这五大组件的原型。
 - ◇ 所有组件都必须运行在“应用程序接口”上，应用程序接口提供了应用程序的入口点、

提供了运行程序的基本框架；

- ✧ 在这里应用程序有两种，一种是普通应用程序，也就是用户自己开发的应用程序，但是该应用程序必须要有本系统中的应用程序接口，否则无法加载配置信息、无法加载组件，也不能运行这些组件实现分布系统的通信；另一种应用程序叫做宿主应用程序，它提供了应用入口点，提供应用程序的基本框架，能够自动加载配置信息，动态创建组件实体，动态运行这些组件，实现应用程序界面的生成，实现人机交互接口，完成应用程序所需的功能（这就是我们以后说的运行器）。
-

- ✧ 提供 C 语言接口、C++语言接口；
- ✧ 使用内置数据对象实现握手、心跳、状态报告、远程日志、远程控制、动态加载、动态卸载、远程配置等功能；
- ✧ 支持 XML 配置，应用程序接口的配置信息由 XML 文件来描述，一个 XML 文件中可以存放多个应用程序接口的配置信息；
- ✧ 使用统一的数据发布/订阅接口，使用该接口函数可以对多种数据进行发布订阅通信；
- ✧ 支持多种数据类型，如短报文、长报文、图像、文件、流式数据(视频、声音)传输；
- ✧ 数据分类处理，支持多线程(线程池)，可以使用多个线程接收处理(内置数据、用户数据、巨型数据、流式数据等)；
- ✧ 支持本地语言数据类型与通信中间件数据类型；
- ✧ 提供多种数据缓冲区(内置数据、用户数据、巨型数据、流式数据等)，每种数据的缓冲区可能不一样，如巨型数据缓冲区与内置数据的缓冲差别相当大；
- ✧ 在同一应用程序中可以支持多种通信中间件或者通信协议(如 DDS、ZMQ、UDP、TCP、数据库等)，支持通信中间件或者通信协议的扩充、扩展和升级；
- ✧ 应用程序业务功能与通信功能相剥离，一个应用程序可在在不同的时刻使用不同的通信中间件进行通信，而不需要更改源程序；
- ✧ 注册数据对象（可以根据名称、类型、域参、发布订阅或者请求应用方式），注销数据对象（当注销对象时，要删除数据对象和相关联的数据存取器以及监听器）；
- ✧ 处理通信中间的内部信息（处理来自各通信中间件的信息，如丢包、发送信息失败、数据过期等）；
- ✧ 提供仿真接口，使得应用程序可以进行数字仿真处理，提供仿真框架、时间调度、算法

性能以及仿真稳定性的统计，仿真的算法、逻辑由客户程序完成；

- ✧ 提供设备接入接口，应用程序可以直接与本地设备通信；
- ✧ 提供日志功能，日志信息可以在本地文件中记录，也可以在本地窗口中显示，还可以在远程运行管控程序中显示，日志信息可以被启用/禁用；
- ✧ 提供 QoS 配置功能，可以使用程控的方式来设置各种通信中间实体的 QoS 策略；
- ✧ 可以获取应用程序以及应用程序内部其它子系统的配置信息、运行状态信息；
- ✧ 可以解析应用程序以及应用程序内部其它子系统的配置信息、运行状态信息。

北京马到科技有限公司

1.5. 建成后的成果

1.5.1. 实现软、硬件互连互通

- ✧ 在作战系统中，将探测设备、数据融合台、指控台、决策台、武器设备，显控台、操作台、发控台等集成在一起，解决了软件、硬件之间的互连互通信；
- ✧ 在装备、武器研制仿真过程中，将计算任务、半实物、显控台、导调、态势、评估等软件集成，进地纯数字仿真、半实物仿真；
- ✧ 在设备测试系统中，可以将被设备、配测设备或者软件、自动化主控台、显控台集成；
- ✧ 在模拟训练系统、教学系统，用于将各种软件、设备、操作台、显控台进行集成；
- ✧ 在电力系统中，可以将各种设备进行集成，采集各设备数据，进行全舰设备监控等。

1.5.2. 提供构件化软件系统框架

- ✧ 在设备测试系统中，可以对配测软件中设备接口、模型和显控进行任意组装，符合被设备测试的测试环境；
- ✧ 在作战系统中，使用构件化思想对软件进行组合，支持作战模式、值班模式和战损模式下软件的组装、部署与运行；
- ✧ 在装备、武器研制仿真过程中，对不同模型的组合，满足不同环境下，仿真试验；
- ✧ 支持种系统软件、硬件的扩充、扩充、升级、更新与维护。

1.5.3. 提供分布式应用系统引擎

- ✧ 提供数字仿真引擎；
- ✧ 提供半实物仿真引擎；
- ✧ 自动化测试引擎；

◇ 模拟训练引擎。

1.5.4. 提供运维管控支持

◇ 提供握手、心跳、状态监视等机制，支持远程监视；

◇ 接受远程指定，支持远程配置、远程控制，实现软件的动态迁移。

2. Adapters(设备接口适配器)

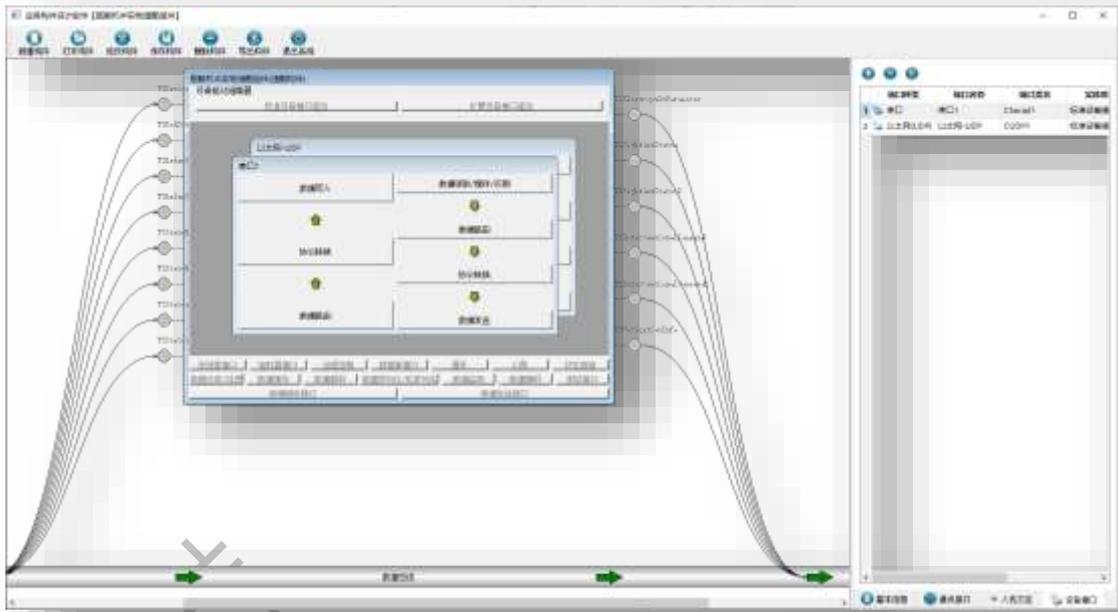
2.1. 概述

设备接口适配器是运行在 MateBus 上的，专门进行物理设备接入的一个可选软件模块。它提供了标准设备接口的驱动程序（如以太网口、串口等）和扩展的设备驱动程序（如 CAN 总线、1553B 总线），提供物理设备接入规范和运行机制，物理设备（传感器、控制器、操作台等）通过指定的设备接口驱动（如以太网口、串口等）与 MateBus 进行通信。

它提供一套可视化设备接入工具，使用该工具可以定制各设备的通信协议，可以定义设备接入计算机的通信接口，可以定义设备数据与通信总线上的数据进行相互转换的映射，可以定义异构系统之间的数据、协议相互转换的映射。

设备通信端口的操作、设备数据的采集以及设备数据与总线数据之间的转换，无需人工开发代码，使用代码生成器生成这些业务代码，供其它系统或者应用程序直接使用。

该适配器可以将各种新老设备接入系统中，与其它系统，软件和硬件互连互通；支持异构系统集成，通过该适配器将不同结构、不同协议的系统免编程的、无缝的集成在一起。这在作战系统、设备测试系统、模拟训练系统、教学系统中会广泛应用。



2.2. 设备接口集成规范

在本系统中，涉及到的物理设备多种多样，其接口也多种多样（如串口、USB 口、以太网口、CAN 总线等），每一种设备接口都有它的访问、配置方法，考虑到各种设备可以自由接入以及新设备的接入，在这里定义了物理设备接口集成规范。该规范提供各种物理设备接口抽象层，通过该抽象接口，使用统一的方法对设备的访问（如数据的读写）和配置。

另外、定义了这些设备存在形式，如使用动态库(.dll)或者共享库(.so)文件的形式存在，使用插件的方式加载，可以动态对设备接入的扩充与扩展。

物理设备接口集成规范规定了：

- 1) 为应用程序或者业务构件提供了物理设备接入时，动态库（共享库）加载、运行、卸载的方法；
- 2) 为各物理设备提供了统一的端口访问（打开、关闭、读取，写入等）方法与配置方法；
- 3) 提供多线程数据接收接口；
- 4) 提供数据高速缓存接口；
- 5) 提供了流式数据处与数据报文理解接口；
- 6) 提供了设备数据识别、设备数据路由、设备数据解码及编码接口；
- 7) 提供了设备数据与总线数据相互转换的接口；

- 8) 提供了数据处理流程。

提供了标准设备接口库：

- 1) 串口设备接口库，通过该接口可以将串口设备接入系统中。
- 2) 以太网设备接口库，通过该接口可以将以太网设备接入系统中。

2.3. 多协议设计工具

在分布式集成平台中，涉及到的应用软件、通信设备多种多样，那么所需要的数据类型也是多种多样，有串口的数据类型、以太网 TCP 协议的数据类型、UDP 报文的数据类型、中间件 DDS 的数据类型、中间件 HLA 的数据类型，CAN 总线的数类型，甚至数据库的表结构等。不同的通信协议，描述方式不一样，描述文件也不一样。

在这里，使用统一的协议设计工具，对各种通信设备接口中的数据类型（即通信协议）进行设计，并使用数据库的对这些数据进行管理。

协议设计工具支持数据的导入，可以将 IDL（接口定义语言）文件中定义的数据导入系统中。

提供数组、序列等复杂数据结构的支持；

提供结构体嵌套支持；

2.4. 代码生成器

可以将协议设计工具中设计的通信协议导出为专用的类型支撑库（如 RTI DDS）；生成的类型支撑库封装了指定通信中间的技术细节，屏蔽了通信中间件的使用，屏蔽了宿主设备、操作系统以及编译器之间的差异。生成的类型支撑库下层能够与分布式集成平台、通信中间件免编程的无缝集成，上层与应用程序、业务构件的无缝集成。

类型支撑库用于对指定的通信中间件（如 RTI DDS）在分布式集成平台中的数据类型扩充、扩展和更新。

2.5. 设备集成设计工具

分布式集成平台提供了各种软件的集成的基本框架和通信平台,为应用程序集成到分布式系统中提供了基础,通信中间件适配器为分布式集成平台通信的多样性提供了实现。设备集成设计工具则是使用可视化工具,实现设备免编程的接入分布式系统中,并实现设备数据与总线数据相互转换,实现异构系统的集成。

设备集成设计工具提供可视化界面,设计各种物理设备接入分布式系统中,并生成指定环境下的代码及代码框架,实现免编程。

设备集成软件我们叫它设备适配器,用于采集设备上的数据,并转换成通信总线上的数据,和接收总线的的数据,转换成设备数据,发送到设备上。

设备集成设计工具的功能有:

- 1) 可以设备适配器分配一个或者多个通信端口,这些通信端口可以是同一类型的(如串口),也可以不同类型的。
- 2) 为每个通信端口分配数据包、消息,并指明发送、接收关系;
- 3) 为设备适配器分配总线数据(如 DDS 主题),并指明发布-订阅关系;
- 4) 设计设备数据到总线数据的转换关系,即从设备接收到数据之后,转换成总线数据的映射关系;
- 5) 设计总线数据到设备数据的转换关系,即从总线接收到数据之后,转换成设备数据的映射关系;
- 6) 各设备的配置信息,如串口(串口号,波特率等)、以太网口(UDP、TCP、IP 地址,端口号)
- 7) 生成指定环境(指定的操作系统、指定的编译器)下的代码及代码框架,并直接或者交差编译成指定的库文件或者可执行文件。

加载这些库文件或者启动可执行文件即可实现设备接入分布式系统,实现可视化免编程的设备接入。

2.6. 建成后的成果

- ✧ 在设备测试系统中用于新旧被测设备的接入；
- ✧ 在设备测试系统中用于新旧配测设备的接入，解决测试人员不会软件开发的问题；
- ✧ 在作战系统中，可以将各中设备接入，实现设备的更新、替换、软件迁移以及新设备加入时，快速执行；
- ✧ 在仿真系统中，可以快速将设备、半实物、传感器、操作台加入系统中；
- ✧ 在电力系统中，可以将种设备集成，执行协议转换，支持设备综合监视、分布式监视以及数据集中采集等。
- ✧ 在模拟训练系统中，可以将设备、传感器、控制器、操作台等接入系统中。

3. MateDDS(数据分发服务)

本 DDS 是 OMG DDS 规范的一个 C++ 的实现。

3.1. 主要特点

- ✧ 符合 OMG DDS 1.4 规范；
- ✧ 符合 OMG RTPS 2.2 规范；
- ✧ 支持 23 种标准 QoS 策略；
- ✧ 支持 Windows、Linux 等主流操作系统；
- ✧ 提供 TCP、UDP、共享内存传输方式；
- ✧ 支持基于 XML 的动态配置，包括数据类型的配置、主题的配置以及 QoS 的配置等；
- ✧ 实时性高，时延低，流量大，在性能上比 ZMQ 优越；
- ✧ 使用标准 C++ 开发，支持 C++11，使用 ASIO 为网络通信库、TinyXml 为 XML 解析器，效率高，可移植好，代码规范，可读性强；

3.2. 同类产品比较

产品名称	产品描述	优点	缺点
RTI DDS	美国 rti 公司开发的一款	上世纪 90 年代研发产品，成	闭源、比较昂贵

	商用 DDS	熟度高、性能优越、工具多，文档齐全，支持多种操作系统，支持多种开发语言。	
OpenDDS	美国密苏里州的 Object Computing 公司开发的开源 DDS	开源，基于 ACE 和 TAO，很容易与 CORBA 集成，有较多的工具	基于 ACE 和 TAO 开发，维护成本非常高的
OpenSplice	AKLINK 开发的一款商用 DDS，并提供部分开源版本	工具、文档较全面，支持多种操作系统、开发语言	商用软件，比较昂贵不支持国产操作系统和硬件设备
AppDDS	中国神州普惠在 OpenDDS 基础上开发的 DDS	可定做开发，支持国产操作系统和硬件设备	闭源，工具及资料少
MateDDS	马到科技在 OMG DDS 标准基础上，使用 c++实现的一个 DDS	使用标准 c++开发，代码规范，可读性、可维护性强，支持国产操作系统和硬件设备	不及商用 DDS 成熟、工具及资料相对缺少

产品名称 产品功能	RTI DDS	OpenSplice	OpenDDS	AppDDS	MateDDS
QoS 支持	标准 22 种 扩展 24 种	标准 22 种	标准 22 种	标准 22 种	标准 22 种
监视工具	rtiddsspy			未知	
Ping 工具	rtiddsping			未知	
实体监视	√			未知	
Excel	√	√	√	未知	
数据记录	√	√		未知	√
路由服务	√			未知	
消息队列	√			未知	
安全	√	√	√	未知	√
RPC				未知	√

3.3. MateDDS 扩展功能

- ◇ 提供数据类型定义与管理工具，使用可视化界面定义数据类型，可以导入、导出 IDL 文件，可以将数据类型直接到出为动态库，实现 DDS 免编程的接入系统。
- ◇ 提供数据监视生成工具，该工具与 RTI DDS 不同的是，RTIDDS 的 rtiddsspy 用于监视指定域的主题发布订阅信息，在命令行里输出；这里的数据监视生成工具，是提供一个可视化的工具，定义要发布订阅的主题，配置 QoS 策略，然后使用代码生

成器，生成指定范围内主题监视的应用程序，生成的应用程序是一个有界面的应用程序，使用结构化来显示（如使用树形窗口显示每个数据），也可以使用曲线、表格、仪表等图形化方式显示指定的数据。

- ◇ 提供数据发送工具，它提供一个可视化的工具，用于定制待发送的数据报文（相当于数据模拟器），可以配置 QoS 策略，然后使用代码生成器应用程序。生成的应用程序可以发送指定范围内的数据，这些数据可以自动地、周期性的发送，也可以手动发送；发送的数据报文内容可以随时间发生变化，也可以固定不变。其它 DDS 不具有此功能。
- ◇ 提供数据库支撑工具，可以根据指定的数据类型映射成关系数据库的表结构，可以根据 DDS 主题的内容对数据库进行插入或者更新操作。
- ◇ 对 JSON 支持，可以将 DDS 各主题内容转换成 JSON，用于与其它语言进行数据交互，或者用于数据的显示。

3.4. 建成后的成果

- ◇ 可以使用本 DDS 进行实时分布式系统集成（包括数字仿真系统、模拟训练系统、自动化测试系统、边缘计算、物联网等）。
- ◇ 支持国产操作系统和硬件设备。

4. Manager(管理器)

管理器是实时分布式系统中一个运维管控工具管理器由以下功能组成，这些工具是一些可以动态加载的插件，根据分布式系统的要求，可以加载不同的插件。



4.1. 远程探测

能够探测分布式系统中未知设备、软件、模块以及通信对象，并对这些对象进行监视和控制。

4.2. 远程监视

对已知系统中分布式环境的监视，包括终端设备、应用程序、模块、通信对象等。

监视对象	描述
分布式节点	<ul style="list-style-type: none"> ● 分布式节点是否已开机
分布式应用程序	<ul style="list-style-type: none"> ● 分布式应用程序是否登录 ● 分布式应用程序是否在线，记录上线时间，最后一次在线时间
应用程序运行状态	<ul style="list-style-type: none"> ● 监视分布式应用程序的运行状态，正常、异常、错误
组件信息	<ul style="list-style-type: none"> ● 组件是否实例化 ● 如果实例化，组件是否正常运行 ● 组件发布订阅的主题信息 ● 如果组件有自己的线程，监视远程组件内各线程的状态 ● 显示组件运行日志 ● 监视数字仿真组件运行时间、同步状态
主题	<ul style="list-style-type: none"> ● 发布主题状态，包括该主题最近发布的时间、累计发布的主题，最近发布失败的时间，发布失败的主题数 ● 订阅主题状态，包括该主题最近订阅成功的时间、累计订阅的主题，最近订阅失败的时间，订阅失败的主题数

日志信息	● 监视远程应用程序、业务组件的日志信息
计算机资源	● CPU、内存、硬盘等硬件或者其它软件的运行情况

4.3. 远程控制

控制对象	描述
分布式应用程序	● 动态加载组件（可用于动态迁移）
	● 动态卸载组件（可用于动态迁移）
	● 重新加载其它应用程序（停止当前应用程序，加载外另的配置 文件，实现加载其它应用程序，典型应用为动态迁移）
	● 控制远程应用程序改变指定的域参与者 QoS
	● 控制远程应用程序改变指定的主题 QoS
组件信息	● 控制远程组件是否发布日志信息
	● 控制远程组件是否发布线程信息
	● 控制远程组件是否发布心跳数据
主题	● 控制发布主题状态信息的发布
	● 控制订阅主题状态信息的发布
	● 远程设置订阅主题的过滤条件
	● 远程设置数据写入器的 QoS
	● 远程设置数据读出器的 QoS
日志信息	● 控制远程的应用程序、组件是否发布日志信息、以及发送频率

4.4. 远程配置

配置对象	描述
域参与者	● QoS 配置
组件	● 通信端口配置
	● 仿真初始化状态数据配置
主题	● 发布主题的 QoS 配置
	● 订阅主题的 QoS 配置
	● 订阅主题的过滤条件配置

4.5. 远程部署

远程部署就是集中管理分布式工程，在进行分布式系统部署时使用远程部署的功能，将分布式系统中各个应用程序资源（业务组件库、配置信息、类型适配器、依赖的资源——如图片、模型、地图、第三方库文件等），按照配置信息，将其部署到远程的计算机（设备）上。

在进行分布式仿真试验和分布式仿真对抗时，需要对作战想定进行部署。

在分布式系统部署之后，可以使用远程部署的远程启动命令来启动每台计算机（设备）上指定的应用程序。

使用远程部署，不需要每台计算机（设备）个逐步安装不同的软件；

使用远程部署，不需要到每台计算机（设备）跟前启动分布式应用程序。

4.6. 远程调试

远程设计主要是提供各种各样的数据源，供系统中其它业务组件调试使用，支持结构化数据类型、嵌套类型，数组类型等；可以使用多用方式进发送（如使用 UDP、DDS、串口等）；支持手动发送（需要数据是手动发送一条）和自动发送（按一般的频率发送）；支持数据类型中各成员值地动态变化。

4.7. 运行管理

运行管理是一些导调程序，不同的系统，可以使用不同的导调程序，主要有以下三种导调程序。

4.7.1. 仿真导调

专用于数字仿真中，协调各仿真组件的仿真运行。

运行管理提供数字仿真所需的仿真同步管理，它能够发布不同频率的仿真步长，控制不同仿真模型的仿真计算。

运行管理使用的时间管理模式是使用墙时钟，这使用得数字仿真与物理实物同步。

运行管理如果要纯数字仿真的情况下，可以使用超实时的方式进行仿真。

4.7.2. 测试导调

4.7.3. 训练导调

4.8. 数据记录

提供多种数据记录的方式

存储方式	描述
一级存储	记录到文件 将数据直接保存到文件中，不进行解码，用于存放实时性高、数据量大的系统
	记录到数据库 将数据直接保存到数据库中，并实时解码，一边接收到数据，一边解码，一边存储数据，用于实时性低、数据量小的系统。
二级存储	先记录到文件再导入数据 将数据直接保存到文件中，不进行解码，在系统运行完成，将文件中的数据进行解码，并导入相应的数据库中，该方式用于实时性高、数据量大与第三方软件集成的系统。

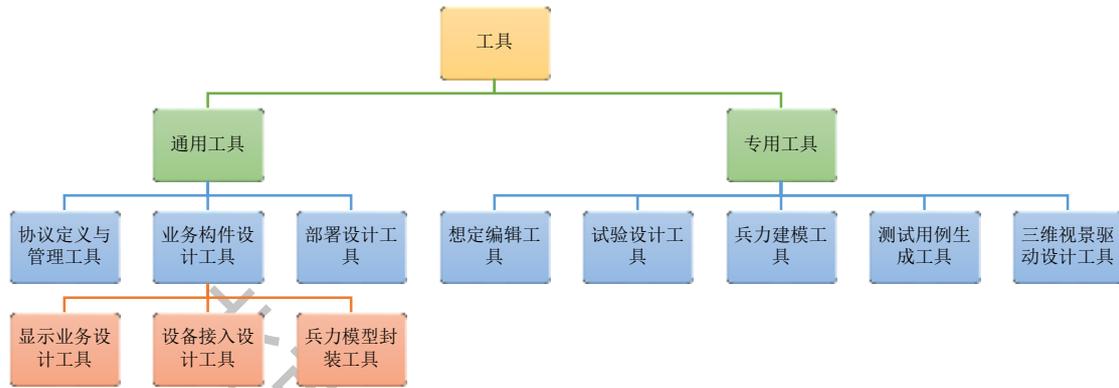
4.9. 数据回放

回放方式	描述
文件回放	直接从文件中取出数据，按时间先后顺序进行回放。
数据库回放	从数据库中取出数据，进行预处理，然后将预处理的结果保存到本地文件中，最后从文件中取出数据按时间先后顺序和指定的要求进行回放。

4.10. 资源调度与管理

(略)

5. Tools(工具)



5.1. 通用工具

通用工具一般用于大多数实时分布式系统的设计，主要有通信协议的设计、业务构件的设计和系统部署的设计。

5.1.1. 协议设计工具

用于设计实时分布式系统用于的数据类型，包括 DDS 数据类型，UDP、TCP、串口等的报文（消息）。

可以从 IDL 文件中导入数据类型。

可以手工录入数据类型。

可以定义 DDS 的主题。

可以导出指定范围的数据类型为动态库（共享库）、XML 文件和 IDL 文件。



5.1.2. 业务设计工具

将应用程序拆分成若干个功能独立的构件（如数据显示构件、设备接入构件、数字仿真构件、计算任务构件）。在这里，每种构件都有一个设计工具与它相对应，都有一个代码生成器与之相对应。设计好业务（包括收发的数据、人机交互、业务处理等）功能之后，可以使用相应的代码生成器，生成可以在 MateBus 上运行的构件库。

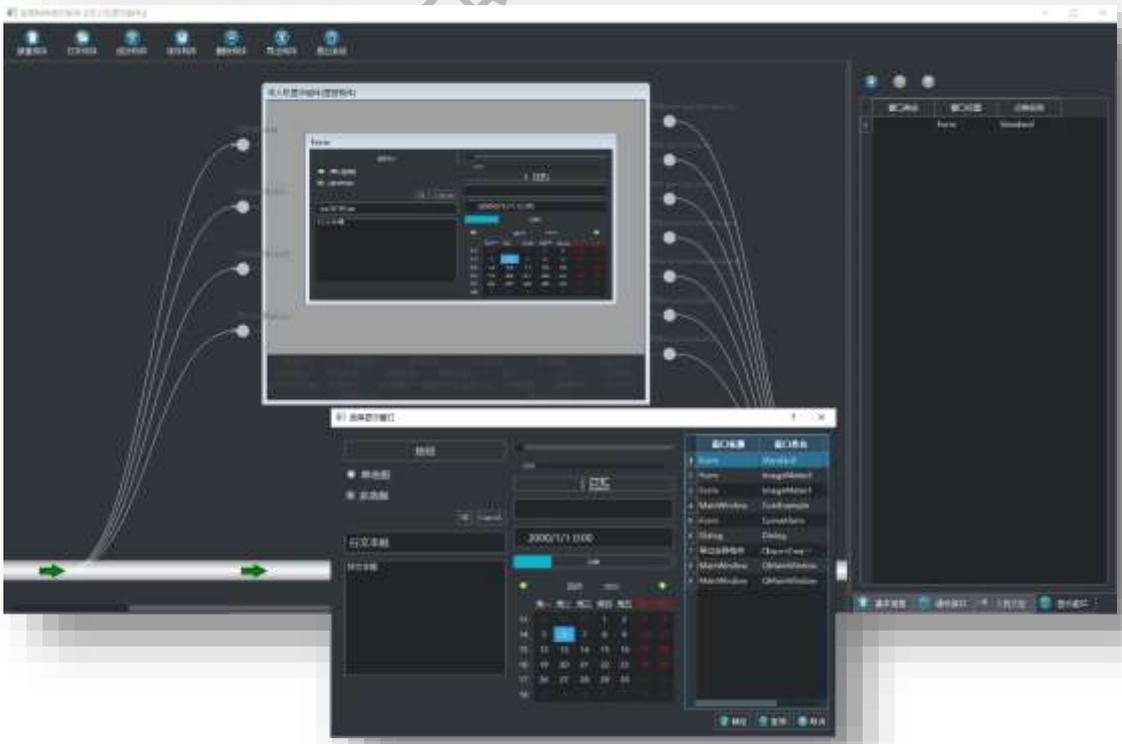
5.1.2.1. 显示业务设计工具

该工具专用于设计显控业务的，这些显控业务可以直接在 MateBus 环境中运行。主要功能包括：对外通信接口的设计（发布-订阅的数据）、人机交互设计、显示界面设计、数据驱动设计、事件响应设计、外观设计等。

支持关系数据管理，具有导入、导出功能，导出的代码可以生成动态连接库，可以在 MateBus 环境中运行。



显控构件对外接口设计



显控构件窗数据驱动与绑定

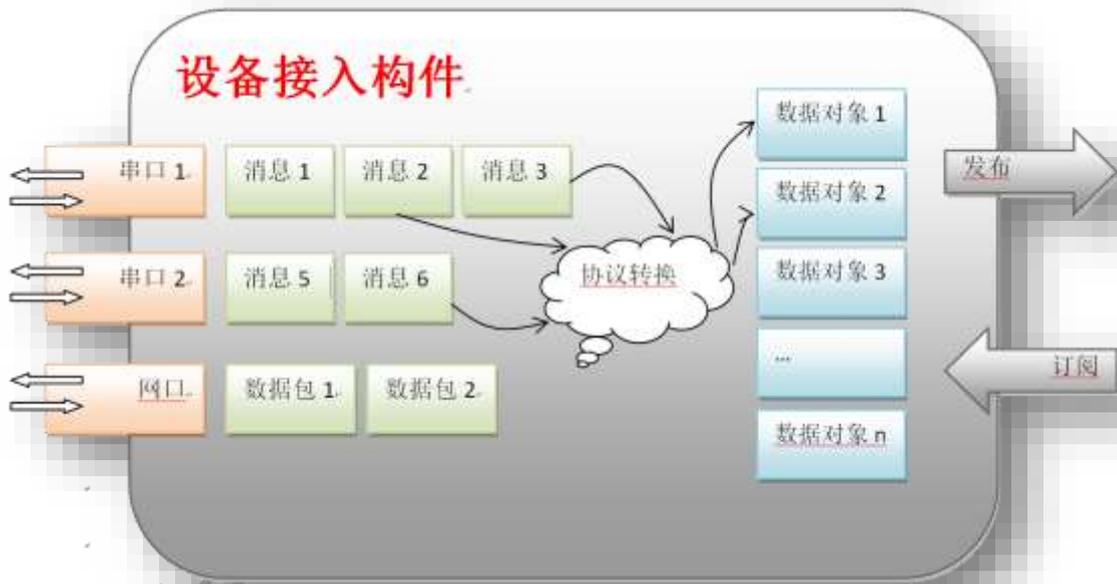


导出显控构件并编译成动态库

5.1.2.2. 适配业务设计工具

适配业务构件就是设备接入构件，它是通过各种通信接口，将物理设备的数据采集下来，然后进行解码，转换成指定的数据对象（如 DDS 的主题，HLA 的对象类、交互类等）进发布，或者将订阅到的数据对象（如 DDS 的主题，HLA 的对象类、交互类），进行协议转换，然后进行编码，生成指定通信接口中的数据包或者消息，将这些数据包或者消息发送到指定的设备。

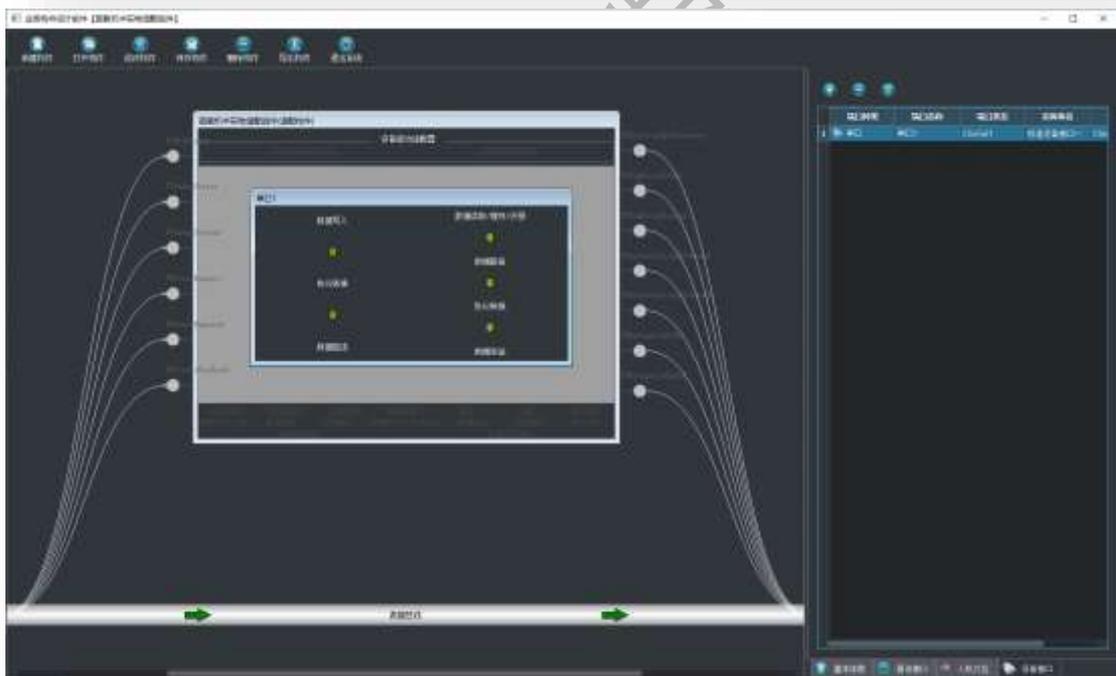
一个设备接入构件可以有一个或者多个通信接口，每个通信接口上可以一种或者多种数据包或者消息。

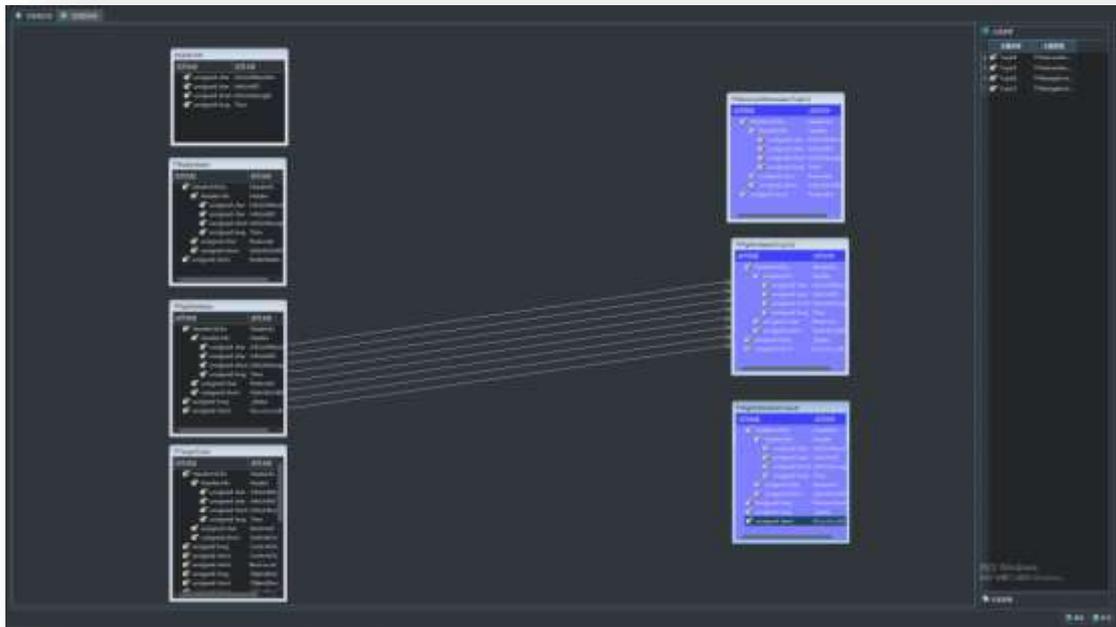


设备接入构件工作原理图

主要功能包括：对外通信接口的设计（发布-订阅的数据）、人机交互设计、设备接口设计、协议转换设计等。

支持关系数据管理，具有导入、导出功能，导出的代码可以生成动态连接库，可以在 MateBus 环境中运行。





协议转换设计

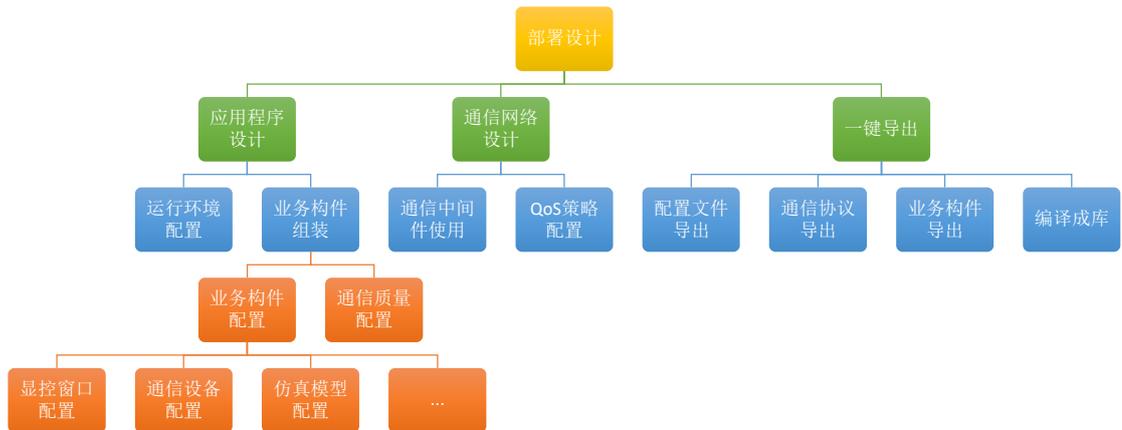
5.1.2.3. 仿真业务设计工具

(略)

5.1.3. 部署设计工具

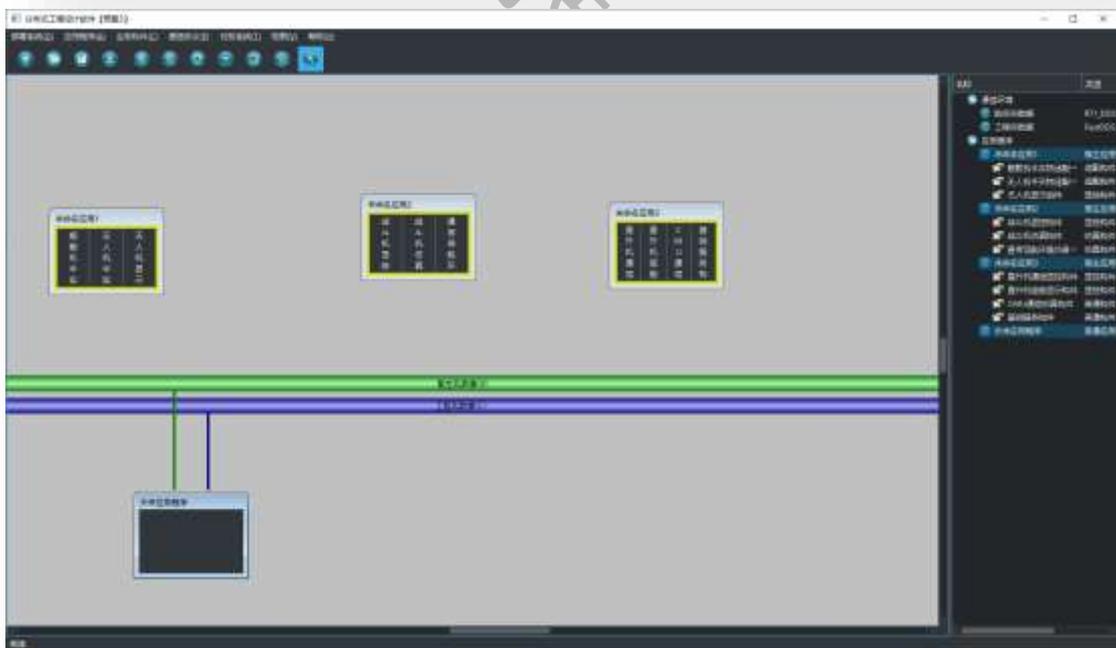
部署设计用于对分布式系统各应用程序的设计,包括应用程序的定义、应用程序组装(由多个业务构件组装成一个分布式应用程序)、应用程序运行环境设计、通信网络设计、通信环境配置等。将设计与配置的结果导出,交给 MateBus 运行环境和相应的框架运行。

部署设计功能组成图如下所示:



5.1.3.1. 定义应用程序

在分布式系统中添加应用程序，这些应用程序可以是外来的，也可本地的，可以普通应用程序，也可是构件化应用程序(应用程序的业务功能由一个或者多个业务构件组装而成)。



5.1.3.1.1. 应用程序基本信息

填写当前应用程序的基本信息,包括所部署的设备,IP 地址、应用程序类型、显示方式、操作系统等。



The screenshot shows a software configuration window titled "应用程序配置窗口" (Application Configuration Window). It has two tabs: "基本信息" (Basic Information) and "构件组装" (Component Assembly). The "基本信息" tab is active, displaying several input fields for configuring an application. The fields and their values are as follows:

Field	Value
应用程序名称 (Application Name)	未命名应用2 (Unnamed Application 2)
IP地址 (IP Address)	192.168.1.159
应用类型 (Application Type)	简单应用 (Simple Application)
界面类型 (Interface Type)	无界面 (No Interface)
终端类型 (Terminal Type)	计算机 (Computer)
显示模式 (Display Mode)	托盘 (Tray)
操作系统 (Operating System)	Windows
多屏显示 (Multi-screen Display)	单屏 (Single Screen)
执行程序 (Executable)	Runner
应用分类 (Application Category)	计算任务 (Calculation Task)

Below these fields is a large text area labeled "应用程序描述" (Application Description), which is currently empty. At the bottom right of the window, there are two buttons: "确定" (OK) and "取消" (Cancel).

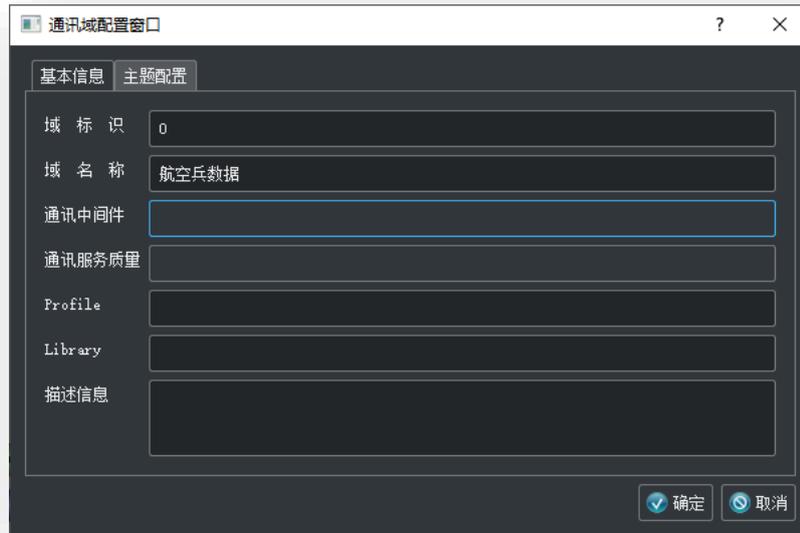
组装业务构件

该功能主要用于构件化应用程序,可以从构件库中选择一个或者多个业务构件组装,生成一个应用程序,由指定的启动器进行运行。这些构件是由前面的业务构件设计工具设计的,存放在数据库中可以使用的构件。



5.1.3.2. 通信配置

设计与配置系统的通信环境。在本系统中,业务与通信解耦合、通信与接口协议解耦合。一个应用程序可以在不同的系统中可以根据需要选择不同的通信中间件或者通信协议。在这里为当前的实时分布工系统中设计使用的通信中间件。如下图所示,通信中间件可以使用 RTIDDS、MateDDS、OpenSplice、OpenDDS、HLA、TENA、UDP、TCP 等。同一分布式系统中可以同时使用多种通信中间件。



5.1.3.3. 系统导出

将系统中各应用程序运行环境、组装信息、通信环境、通信热量要求、业务配置、接口协议等进行一键导出。

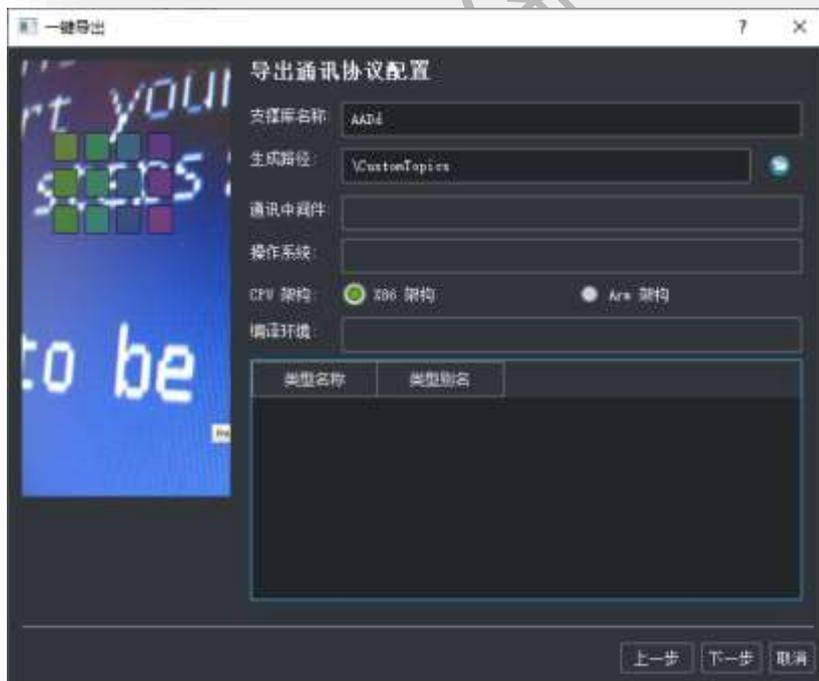
导出配置信息

将系统的配置信息导出为 XML 文件，可实际的应用程序（如本地应用程序、运行器、管理器、MateBus 等）使用。这些应用程序根据配置信息可以加载指定的构件库、使用指定的接口协议、通信设备、显控窗口；可以根据这些配置信息进行数据记录、数据解码、数据编码、数据显示等）。



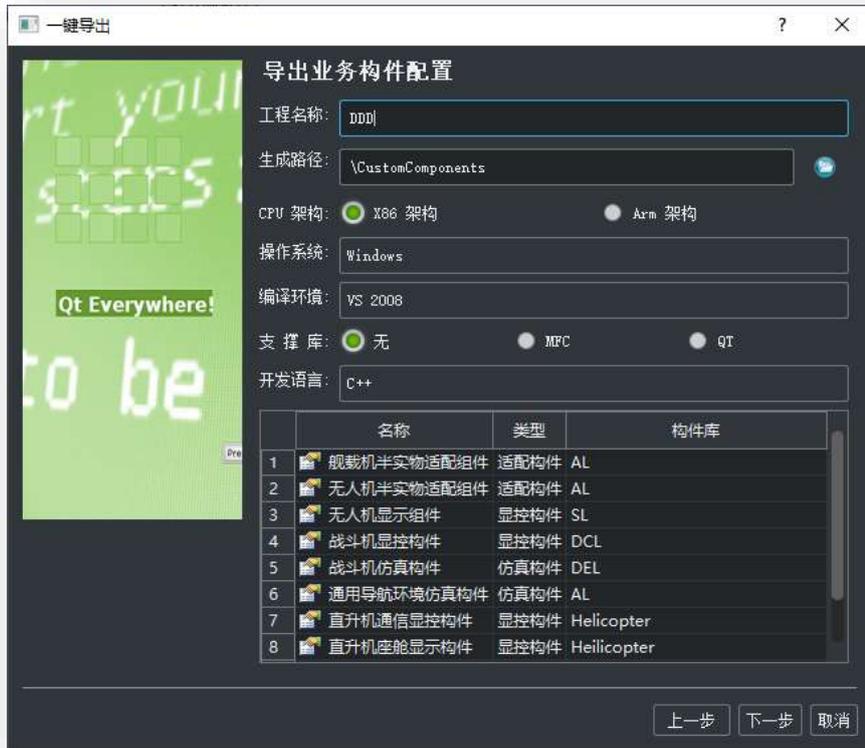
导出接口协议

导出接口协议就是将系统中用到的数据类型、数据包、消息、报文等信息导出为指定通信中间件的类型适配器，由 MateBus 加载使用，支撑系统的运行。



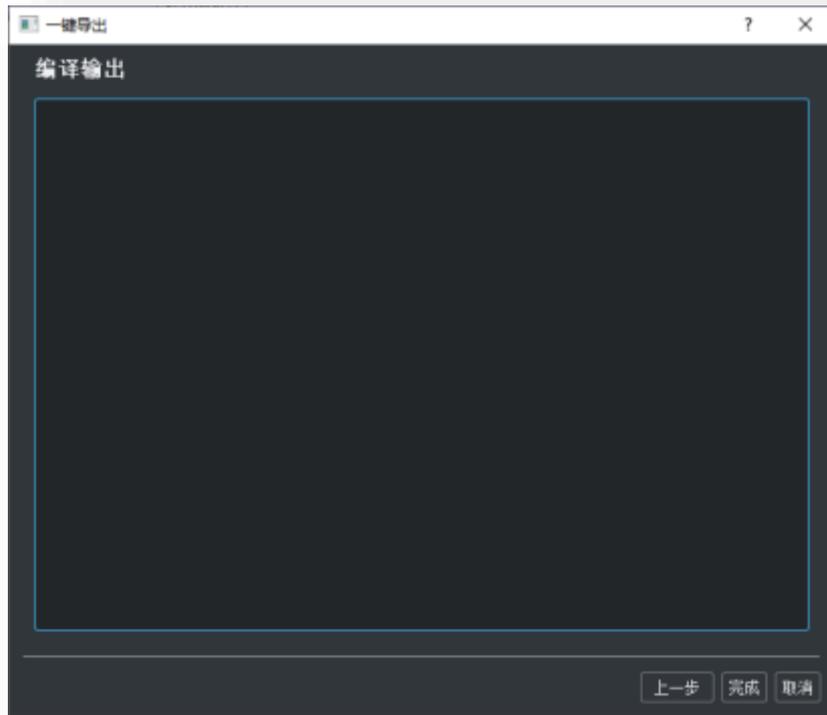
导出业务构件

将系统中各应用程序使用到业务构件（这些业务构件在数据库中只是一个描述信息，并不是一起可以直接使用动态库）导出为代码，生成可以动态加载的构件库。



编译成动态库

将导出的接口协议和业务构件的动态库代码在指定的环境下进行编译，生成系统可以直接运行的动态库。



5.2. 专用工具

专用工具是用于某个行业的，如仿真，模拟训练，测试、监控等。这里定义一些工具标准和规范，支持数据库，支持代码生成，支持导入导出，生成的代码可以无缝集成到 MateBus 的分布式系统中。

(略)

6. Cases(案例)

6.1. XXX 作战原型实验室建设

6.1.1. 系统概述

本系统使用数字仿真的方法研究大型船舶作战原理系统。系统中各数字模型在分布式仿真环境是运行，它们使用构件化方法进行设计，在 MateBus 环境中运行。系统除了提供运行环境之外，还提供了显控集成框架，确保系统同几十台显控台显示风格统一；另外还提供

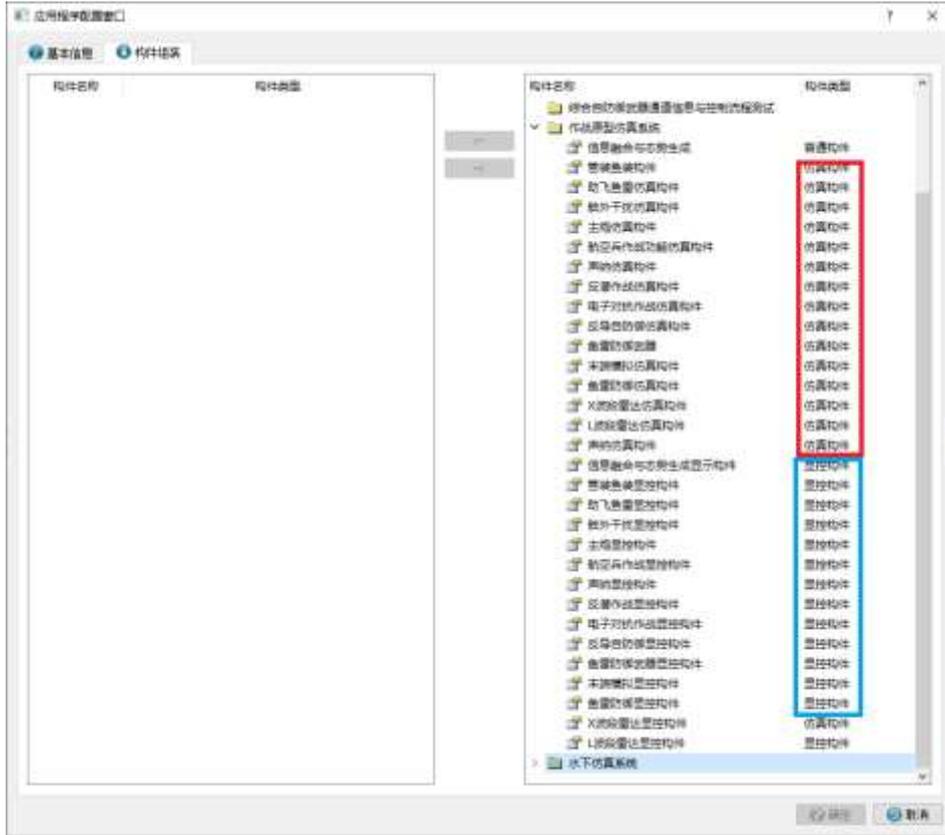
运维管控功能，对这些数字仿真模型进行远程部署、远程启动、远程控制、远程监视、仿真导调等。

6.1.2. 系统组成



6.1.3. 系统实现

- ◇ 系统中使用 RTIDDS 通信中间件。
- ◇ 使用本系统提供的协议设计工具，设计出本系统中几十个台位中使用的接口协议。然后使用本系统的协议导出工具，生成 DDS 的代码，实现各台位之间的通信。
- ◇ 使用本系统提供业务构件设计工具，设计出各显控构件、设备接入构件、数字仿真构件等，并用相应的代码生成器，生成代码框架、代码工程并进行编译。
- ◇ 使用手工编程方法完善数字仿真构件，完成各数字仿真构件的业务功能。
- ◇ 使用本系统提供的部署设计工具对系统进行部署设计，应用程序设计、组装与配置，生成系统的配置文件。
- ◇ 由系统提供的管理器对系统进行远程部署，然后远程启动、远程控制启动系统，实现系统的仿真。由系统提供的记录工具实时记录仿真产生的实时数据，供事后处理使用。



6.1.4. Mate 在系统中的应用

使用接口协议统一设计与代码生成的方法，确保系统中各仿真模型、显控台使用的协议高度一致性，大幅度的提高了系统联调效率、降低了系统难度、减少系统的工作量。

使用统一建模方法对仿真模型、显控台、操控台进行建模，并使用相应的代码生成器生成各模型、显控台、操控台的业务代码和代码框架，简化了系统的设计、提高了系统可靠性、降低了系统的难度、减少了系统的工作量。

提供了显控集成框架，使用几十家设备单位的显控台显示风格一致，显控台与显控台、显控台与仿真模型、显控台与操控台之间互联互通。

提供了仿真引擎及仿真运行框架，使得数字仿真模型和半实物模型进行同步、协同仿真。

提供了数据总线、集成了 DDS，屏蔽了系统的使用细节和技术难点。

6.2. XXX 作战协同实验室建设

(略)

6.2.1. 系统组成



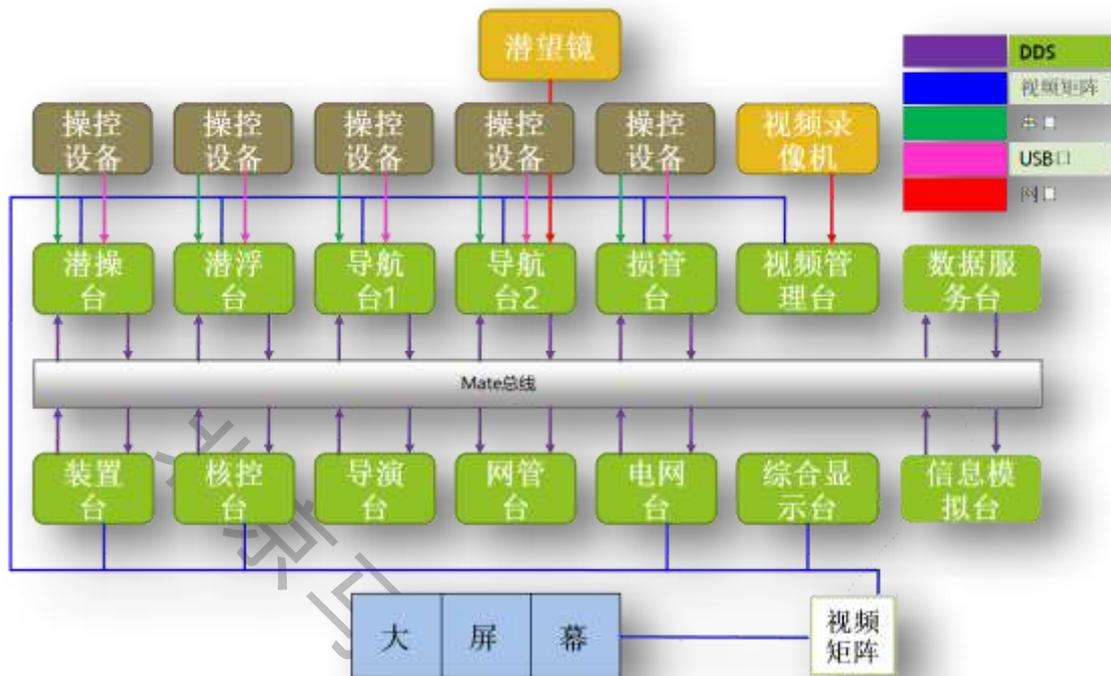
6.2.2. Mate 总线在系统中的应用

6.3. XXX 水下仿真系统

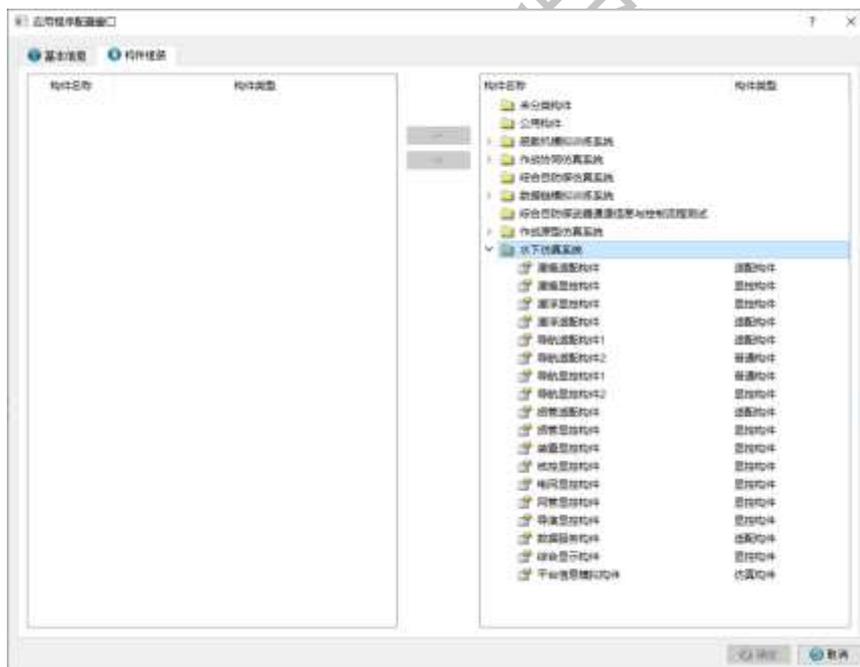
6.3.1. 系统概述

本系统是一个半实物仿真系统，它由一些数字仿真模型、显控台和操控台组成，通过 MateBus 将各台位及相关硬件、软件连接在一起，实现半实物仿真。

6.3.2. 系统组成



6.3.3. 系统实现



6.3.4. Mate 总线项目中的应用

6.4. XXX 构件化战场仿真系统

构件化战场仿真系统旨在以构件化形式建立一套战场仿真推演平台，并将现有战场对抗仿真系统中各仿真软件（包括雷达、声呐、导航、仿真导演、外部信息模拟器、导弹武器模拟器、鱼雷声抗模拟器、光电桅杆模拟器等 12 套仿真软件）以构件化形式集成到该系统中，项目的建设目标如下：

- 1) 基于 DDS 总线型和发布订阅的通信机制，实现系统中各仿真软件互联互通；
- 2) 系统中各仿真模块可分别运行在不同的平台上，实现跨平台的数据交互；
- 3) 以构件化形式将仿真软件各功能模块独立封装，实现各软件在平台上的即插即用功能，方便软件的维护和升级；
- 4) 将仿真软件显控界面与业务模块、通信模块进行解耦，实现显控界面的动态部署；
- 5) 提供统一的运行管控工具，用于控制各仿真节点的启动、暂停并监控各仿真节点的运行状态，方便系统远程调试。

6.4.1. 系统架构

构件化软件设计采用分层设计的思想，使得系统具有良好的层次性，简化开发过程，增强可维护性，同时提高了系统的互操作性、可扩展性。为了满足构件化软件集成需求，本期将各个软件模块从界面层、业务层和通信层三个层次进行改造，各软件改造后的体系架构如图 6-1 所示：

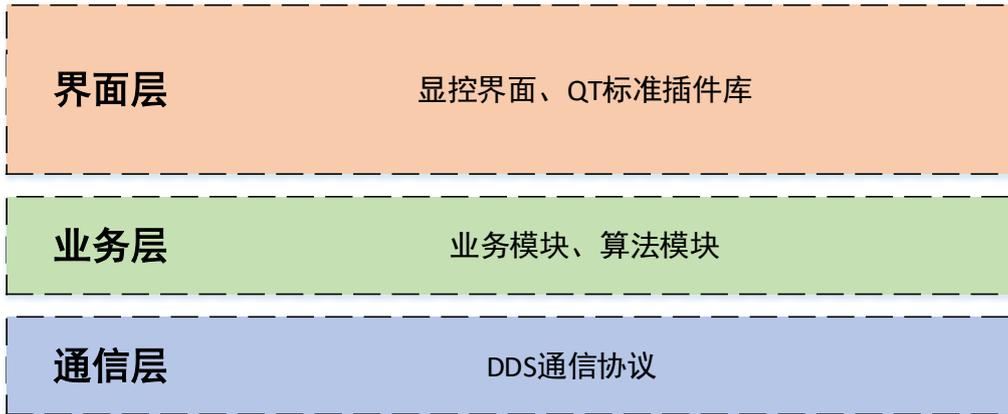


图 6-1 构件化软件体系架构图

- **界面层**：使用 Qt 作为界面开发工具，提供 Qt 标准插件库，基于原软件界面元素，重新开发新的显控界面；
- **业务层**：将原软件界面模块、业务模块和通信模块进行解耦，从中剥离出业务模块和算法模块并分别进行独立封装，以构件化形式重新改造业务代码，生成新的应用程序；
- **通信层**：在改造后的应用程序底层封装基于 DDS 标准的通信协议，以发布订阅机制完成数据交互。

6.4.2. 系统部署

构件化战场仿真系统部署如图所示：

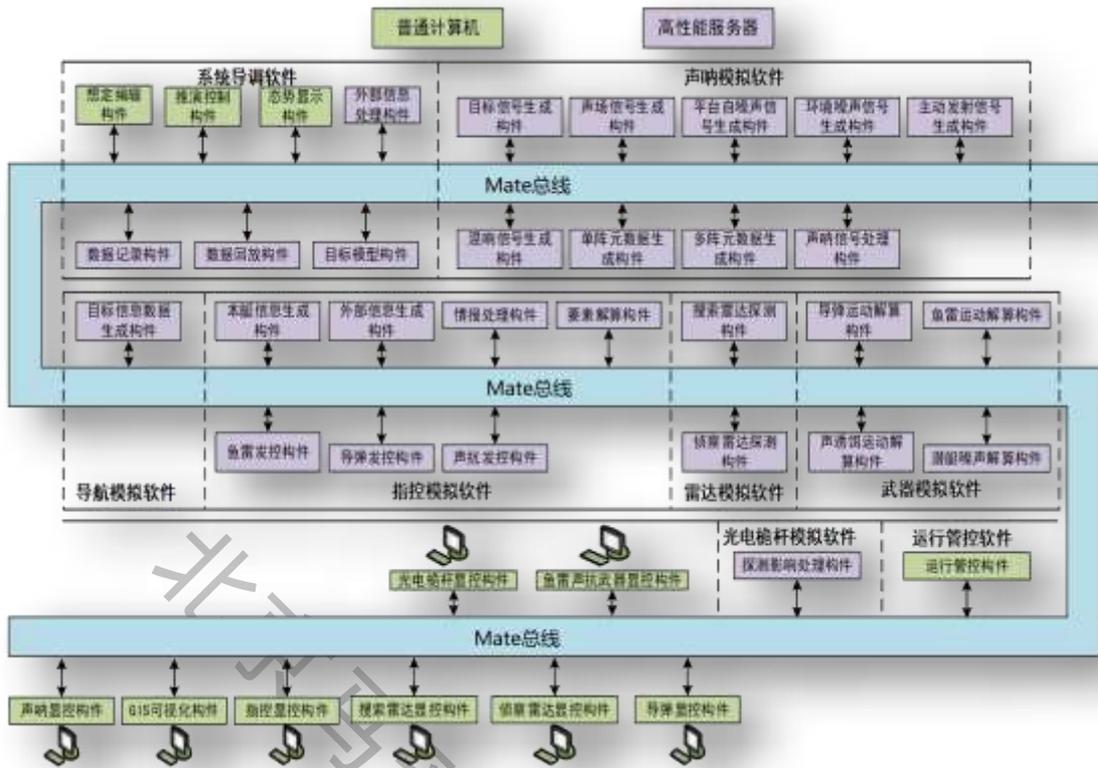


构件化战场仿真系统部署示意图

图中显控构件部署在显控计算机上，主要负责各软件单元的显控界面，可外接多台显示器分别显示不同软件的界面；其它业务处理构件部署于高性能刀片服务器上用于各业务数据的计算；各构件之间均通过 DDS 发布/订阅关系进行数据交互。

6.4.3. 系统构件框图

构件化战场仿真系统各构件间交互关系如图所示：



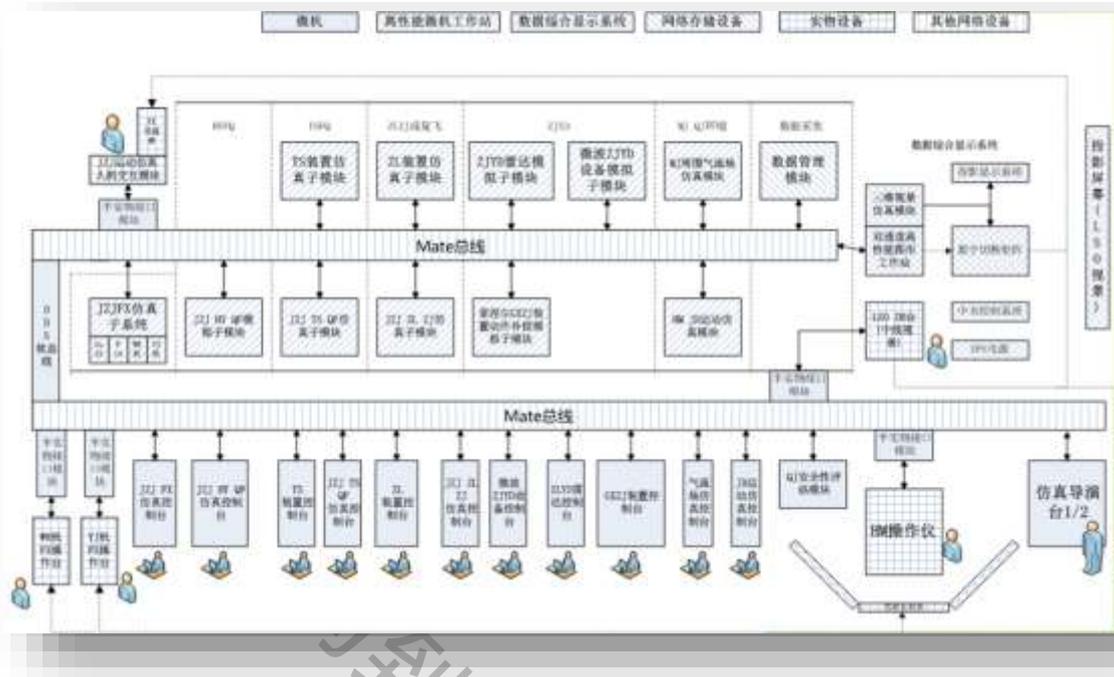
软件框图

6.5. XXX 舰载机起降模拟训练系统

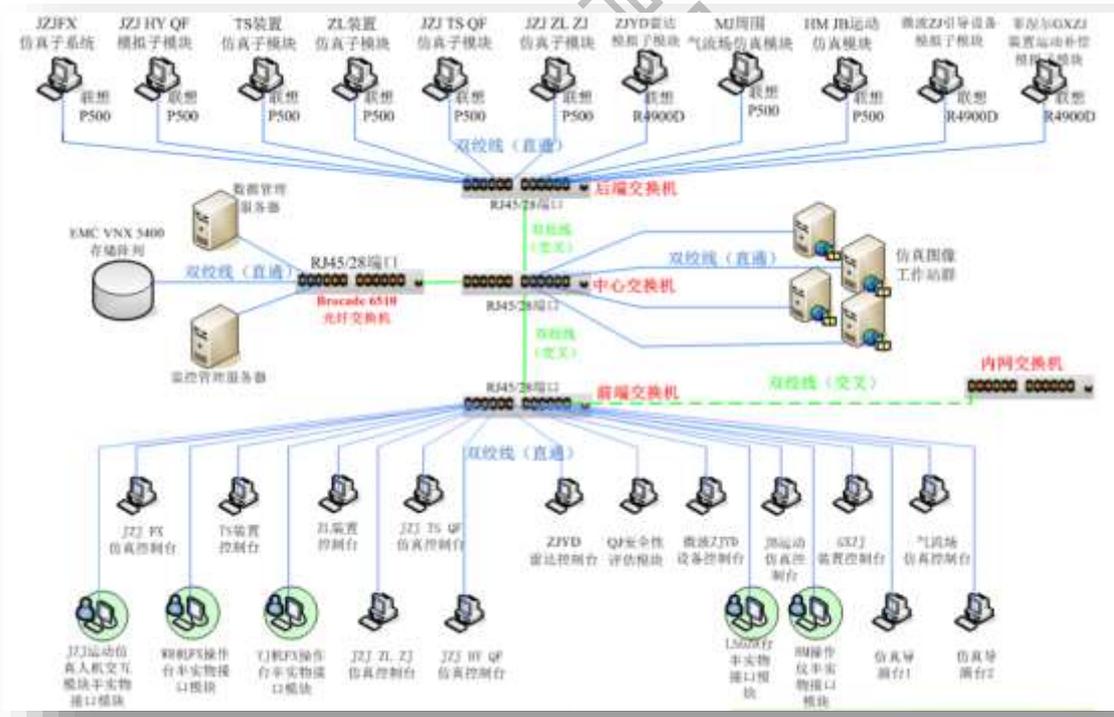
本系统是一个模拟训练系统，使用仿真技术对舰载机、航母等进行模拟，使用半实物座舱模型舰载机的操作。

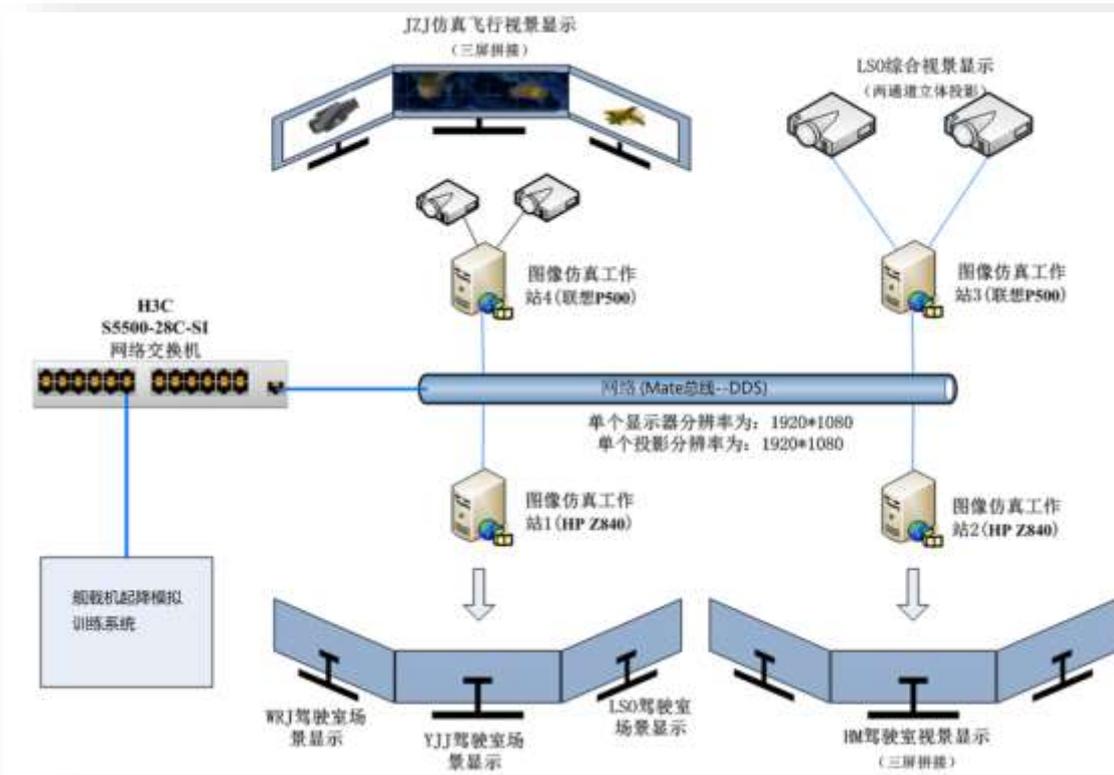
本系统主要用于舰载机在航母上进行滑跃起飞、弹射起飞、微波引导着陆等多个训练科目。

6.5.1. 系统组成



6.5.2. 系统部署



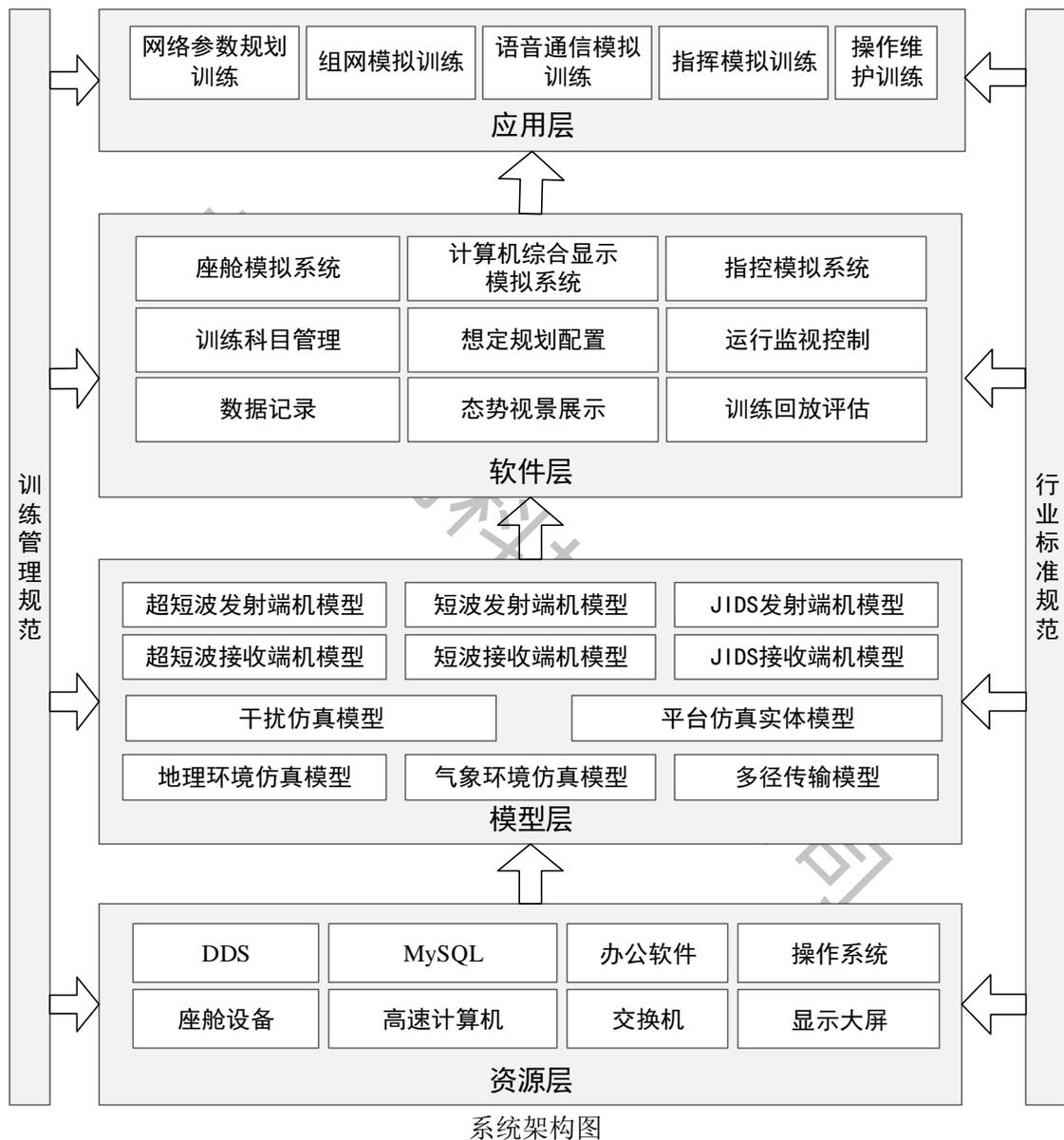


科
技
有
限
公
司

6.5.4. Mate 总线在系统中的应用

6.6. XXX 通信数据链模拟训练系统

6.6.1. 系统架构



1. 应用层：支持陆航数据链的网络规划训练、数据链组网训练、语音通信训练、空地指挥训练、装备操作与维护训练等典型应用；

2. 软件层：包括座舱子系统、计算机综合显示模拟子系统、指控模拟子系统和导调监控模拟子系统（科目管理、想定规划、训练控制监视、态势视景展示、

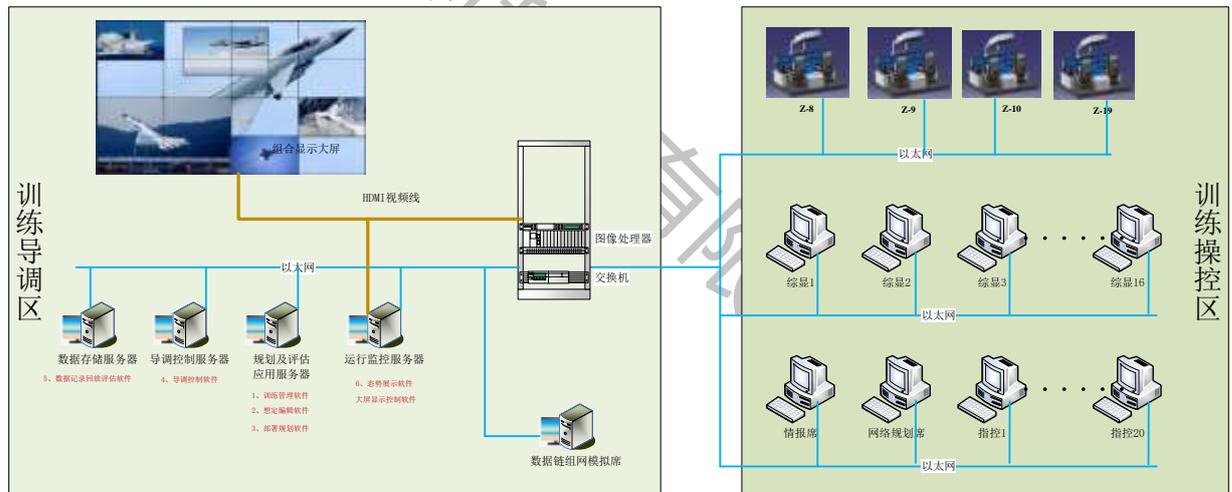
训练数据记录回放评估)；

3. 模型层：包括超短波发射端机模型、超短波接收端机模型、短波发射端机模型、短波接收端机模型、JIDS 发射端机模型、JIDS 接收端机模型、平台仿真实体模型、地理环境仿真模型、气象环境仿真模型、干扰仿真模型、多径传输模型等设计；

4. 资源层：主要包括 DDS、MySQL、办公软件、显示设备、高速计算机、交换机、操作系统，网络连接设备等一系列仿真资源。

6.6.2. 系统布局

战术数据链与通信网络训练模拟系统由导调监控分系统、数据链组网模拟分系统、直升机座舱模拟分系统、计算机综合显示模拟分系统和指控分系统五个分系统组成，各个分系统通过以太网进行数据交互，分系统内部的各个设备也是基于以太网进行通信，为了提高数据交互的效率和仿真的一致性，建议交换机采用千兆以太网卡。系统布局如下图所示



系统布局图

6.6.3. Mate 总线在系统中的应用

6.7. 自动化测试系统

6.7.1. 测试系统功能

6.7.1.1. 系统对接联调

该功能一般用于总体所对系统进行接口一致性联调。

系统对接联调，就是对指定的系统（如大型船舶中的作战系统）中各设备之间进行两两接口协议的联调，确保各设备中各报文的正确性。

6.7.1.2. 通道联调试验

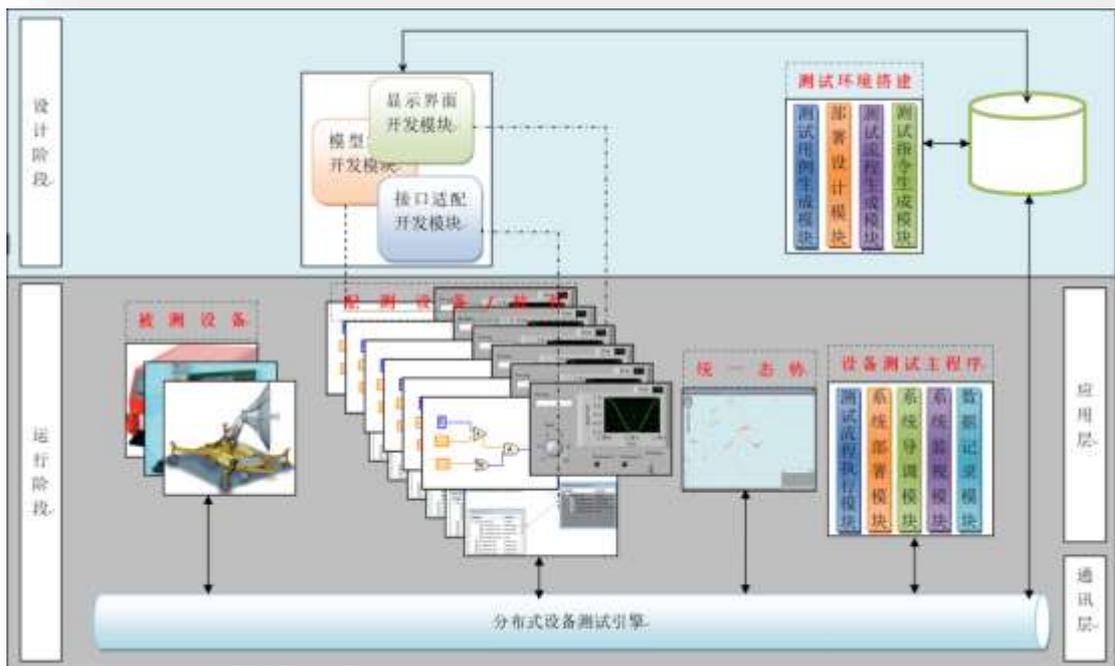
该功能用于某个通信进行联调，如大型船舶作战系统中，对海对陆打击、综合自防御、反潜等通道进行测试，检测这些数据流程（数据从目标、到传感器、到指控、到武器等）是否完整、功能、性能是否达标，验证各通道功能的完好性。

6.7.1.3. 综合任务联动

综合任务联动，是在通道联调的基础之上，再做出对系统功能进行测

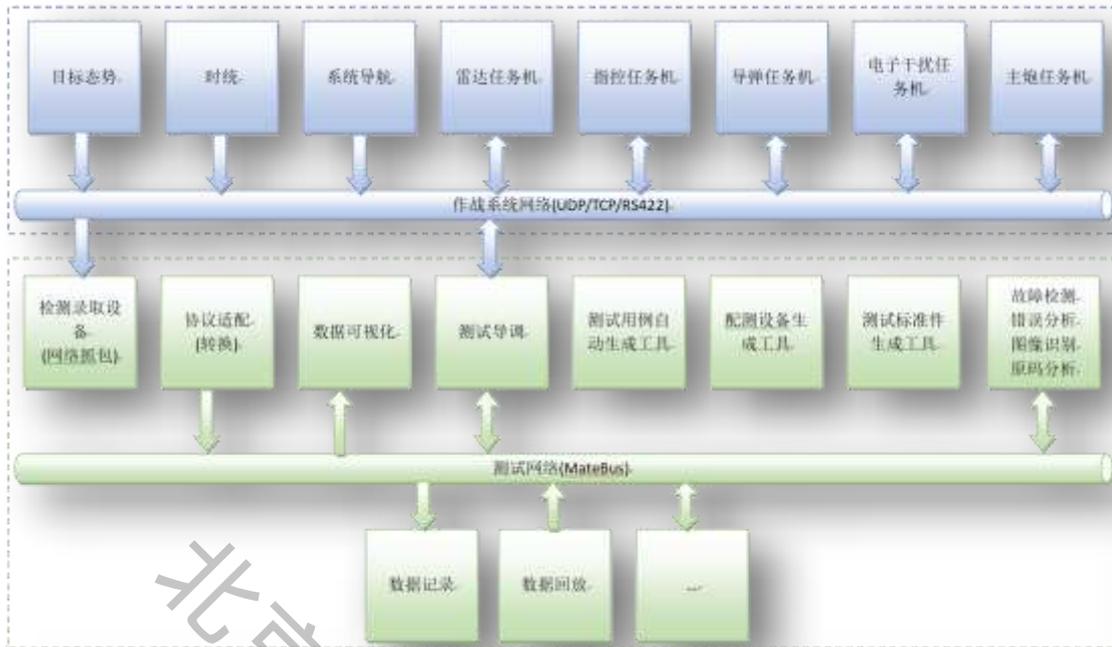
6.7.2. 系统组成

系统由两大部分组成，一部分为自动化测试系统的设计，它是由一系列的工具有组成（ICD 软件、测试用例生成工具、测试科目测试流速定做工具、系统部署工具、测试指令生成工具、测试界面开发工具、协议转换代码生成工具、接口适配代码生成工具等）；另一部分为运行时环境（包含测试仿真环境、分布测试引擎）。



6.7.3. 系统应用

如上图所示，为船舶作战系统测试的应用



6.7.4. Mate 总线在系统中的应用

北京马列科技有限公司