

# 说明书

---

## 基于 LSOA 构架的开放式控制方法

### 技术领域

本发明涉及一种开放式结构控制方法，具体涉及一种基于 LSOA（Lightweight Service-Oriented Architectures，轻量级 SOA）构架的开放式控制方法。

### 背景技术

控制系统技术正经历着从传统封闭式向开放式控制系统发展的过程。开放式控制系统是机械制造技术领域的革命性飞跃，其硬件、软件和总线规范都是标准的，对外开放的，使控制系统满足模块化、标准化和网络化的要求，因此开放式控制系统被公认为将取代传统封闭式控制系统。目前采用一般架构的开放式控制系统，也具有一定的可扩充性和可定制性，达到开放，通用的基本要求；但是其配置过程比较繁琐，缺乏良好的可交互性以及友好的人机界面，而且不能在系统运行时动态配置系统的运行流程。

据查，上海广茂达伙伴机器人有限公司申请了“基于 USB 总线的开放式运动控制系统”专利（申请号：200610148131.X），该系统在通用的 Windows 平台上实现实时运动控制算法，可以进行两层架构和一层架构控制的无缝切换。但在体系结构方面存在通用性上的不足：首先，它基于特定的硬件平台研制，不能跨越硬件平台移植；其次控制软件无法对不同的控制对象上通用；在体系结构方面缺乏标准化规范，在不同厂家的硬件产品和控制软件之间不通用。

上海电气（集团）总公司研究中心申请了“开放式数字控制器”专利（专利号：03210641.6），该数字控制器采用通用工业控制微机的主板，配以标准的总线接口板或通讯接口板，其基本结构具有一定通用性，但需更换通讯接口板及相应控制软件，其通用性有很大局限性。

中国科学院自动化研究所申请了“开放式工业机器人控制平台”的专利（申请号：00129523.3）。该专利中开放式工业机器人控制平台属于通用的工业机器人控制系统。其突出特点是将通用的工业计算机引入传统的机器人控制系统中，并采用独特的开放式及模块化结构，将原来结构封闭的控制器扩展为结构开放的控制平台。这一发明主要应用在基于工业计算机平台上的机器人，因而无法推广到一般硬件平台上的机器人控制。

上海交通大学申请的“面向对象的机器人通用开放控制系统”及在此基础上申请的“开放式网络机器人通用控制系统”专利（申请号：03128900.2），采用把各部件封装成功能模块，

通过约束各模块间的交互和规范各模块所属层次，实现通用的控制体系。“开放式网络机器人通用控制系统”通过服务公布一查找模型实现了机器人和机器人设备间的自发互联和机器人组件的网络即插即用。但这两专利系统采用模块化结构，这种紧耦合的模块化结构选择特定的上层控制软件作为其模块化的参考依据，使得这些模块在不同应用时必须重新编程或修改，影响了通用性。

清华大学申请的“开放式结构机器人控制器”专利（申请号：03100077.0），其应用软件系统 RTOC 主要由三个运行层次组成：基础层是在各个特殊的设备驱动程序之上，抽象出设备无关的访问接口；中间层负责管理系统伺服周期的定义和运行；应用层为高层应用程序，可监控系统状态。这三个层次能够各自独立运行，在实现各自功能的基础上，每一层为其必要的用户层提供必要的访问接口。但它也采用紧耦合的模块化结构，必然存在通用性及灵活性方面的问题。

另外，欧美及日本等工业发达国家都相继在开放式结构控制系统方面进行了大量的投入和研究，其中最具代表性和影响的相关研究计划有：1) 美国的 NGC 和 OMAC 计划；2) 欧盟的 OSACA 计划和 OCEAN 计划；3) 日本的 OSEC 计划。

综上所述，现有的几种开放式控制技术方案的，可以替换具体的控制算法模块以适应不同需求。但是都存在以下三个问题：（1）紧耦合的模块化结构在不同应用时必须重新编写或修改功能模块，并且各功能模块及其处理流程在系统启动后就已加载，不能在线维护更新，通用性、灵活性较差；（2）只能适应线性的控制流程，当各种模块间关系比较复杂时，无法建立起与之适应的复杂控制流程；（3）控制流程的配置方式较复杂，缺乏灵活性，交互性差，无法在系统运行时动态修改和配置控制流程。

## 发明内容

本发明的目的在于克服现有控制技术方案的封闭结构，系统功能固定、不具有柔性、兼容性差，维护费用昂贵等缺陷，提出基于 LSOA 架构的开放式控制方法，其基本思想是以服务为核心，将各功能组件提供的资源整合成可操作的、基于标准的服务，使其能被重新组合和应用，并且设计了一个可在线配置业务流程的软总线，是一个具有开放式系统结构、可扩展、可重构、可在线升级维护、应用服务容错能力强等特点的控制方法。

本发明的目的通过下述技术方案实现：基于 LSOA 构架的开放式控制方法，其特征在于，包括以下步骤：

- S1、应用服务层向总线控制层提交注册服务的请求；
- S2、向总线控制层配置服务流程；
- S3、向总线控制层发送流程请求消息；

- S4、总线控制层根据流程请求消息的优先级对流程请求进行调度；
- S5、总线控制层对流程请求进行处理，然后向相应服务发送流程请求；
- S6、应用服务收到请求消息，执行完指定的控制逻辑后向总线控制层返回结果；
- S7、总线控制层将结果返回。

步骤 S2 所述配置服务流程包括以下步骤：

- S21、流程配置程序从总线内核的数据库获取所有服务的相关信息列表；
- S22、选取一个服务当中提供的某个功能，将其作为新的流程节点添加到服务流程列表中；
- S23、流程配置程序将根据步骤 S22 中所选取的流程节点的输出参数类型，更新可添加的服务流程信息列表；
- S24、如果需要添加新的服务，则返回步骤 S22 继续添加；否则完成服务流程配置；
- S25、为刚配置好的服务流程生成服务流程 ID；
- S26、步骤S25的操作成功后，则将配置好的服务流程信息保存在总线内核的数据库中。

步骤S5所述流程为：请求信息处理流程、注册信息处理流程、状态信息处理流程或总线同步信息处理流程。

当所述流程为注册信息处理流程时，注册信息处理流程包括以下步骤：

- S511、总线内核接收到请求消息；
- S512、总线内核对请求消息进行分析，判断是否注册请求消息，是则到步骤 S513，否则执行其它请求处理流程；
- S513、总线内核开启线程处理注册请求；
- S514、总线内核为服务分配服务 ID，这个 ID 是本总线内核实例范围中唯一的；
- S515、总线内核将服务信息写入总线内核数据库，并将服务ID发送给请求注册的服务。

当所述流程为请求信息处理流程时，请求信息处理流程包括以下步骤：

- S531、监听流程请求队列，当存在请求信息待处理时，转步骤 S532；
- S532、查找请求所需调用的服务流程，若查找成功则转 S533，否则向服务请求端发送调用服务失败信息，从消息队列中删除该请求；
- S533、按总线提供的流程，调用相应的服务；
- S534、处理完请求信息后，从请求队列中删除该请求信息，重新进入等待状态。

当所述流程为状态信息处理流程时，状态信息处理流程包括以下步骤：

- S541、总线内核监听状态信息队列，当存在状态信息待处理时转步骤 S542；
- S542、分析状态信息，将内含的服务状态信息和服务连接状态信息更新到总线内核维护的

状态信息中；

S543、判断服务状态是否出现异常，是则转步骤 S544，否则转步骤 S546；

S544、检查是否有替换服务。替换服务是在流程定义的时候设置的，每一个流程节点都可以设置一个或多个的替换服务方法。若存在替换服务则转步骤 S545，若不存在替换服务则通知系统中的各个服务停止运行与该出现异常的服务相关的程序；

S545、将替换了的服务信息在总线数据库中进行更新。

本发明相对于现有技术具有如下的优点及效果：

1、以 SOA 体系结构为基础，具有 SOA 体系所具有的开放性，通用性，可维护性等重要特点；并且设计了软总线控制平台，可方便地配置控制流程，增强了系统的灵活性，交互性，以及在线升级能力。

2、能够方便的整合不同的设备，减少为了调试和维护而装配系统所消耗的成本。

3、可在线维护各控制软件及其相应的处理流程，无需重启系统，有更强的通用性和灵活性。

#### 附图说明

图 1 是基于 LSOA 构架的开放式控制系统的体系架构图；

图 2；

图 3 图；

图 4；

图 5；

图 6图；

图 7 框图；

图 8图；

图 9图；

图 2图；

图 3 图。

#### 具体实施方式

下面结合实施例及附图对本发明作进一步详细的描述，但本发明的实施方式不限于此。

实施例

如图 1 所示，本发明方法所基于的 LSOA 架构的开放式控制系统，主要包括六个层次：应用服务层、服务支持层、传输层、总线控制层、操作系统层和设备层。总线控制层是最核心的，其功能模块可划分为信息中心、消息处理模块、调度模块、安全管理模块、容错模块和业务流程配置模块，如图 2 所示。总线控制层负责处理整个控制系统各个模块之间的数据交换和相互调用，以及整个闭环控制的执行流程；各种数据流，包括状态，请求，响应等信息都需要由总线控制层接收，再由总线控制层根据一定规则进行处理或者转发到相关的服务进行处理。

为了实现系统中各个层次之间的松耦合关系，本系统采用基于消息传输的交互机制。也就是说在总线内核和应用服务之间的交互，以及总线内核和设备服务之间的交互，都是通过消息传递来完成的，和总线以及服务的实现方式无关，这样就使得整个系统的可维护性大大提高。

而系统所采用的嵌入式数据库，是基于服务的软总线控制平台下的。该数据库与 C/S 模式的数据库不同，是直接嵌入到控制总线中的；访问该数据库不像访问一般的 C/S 模式的数据库那样，要先启动服务端，而只需直接调用该数据库中提供的 API 即可。这样就消除了访问数据库之前要先启动服务端并与服务端建立连接的开销。该数据库只提供编程语言级的 API，而不支持 SQL 和事务机制。这样的好处在于：（1）执行更加快速，因为不需要对 SQL 进行解释编译也不需要执行事务的管理；（2）可靠性更高，因为没有 SQL 和事务机制而只有少量的部分在数据库中运行。该数据库提供记录级的单粒度封锁机制，最大限度地提高了系统的并发度。在数据恢复方面，该数据库采用完全数据备份和日志备份相结合的方式，在故障发生后，可以有效地恢复数据，确保数据的正确性和完整性。

本发明方法所基于的 LSOA 架构的开放式控制系统，在一个较佳的实施例，可以采用六自由度机器人来实现。如图 3 所示，应用于六自由度机器人的实施例，包括由控制 PC 计算机和与其相连的周边设备（包括传感器、机械臂）所组成的硬件部分，以及对该硬件部分进行控制的操作系统和应用服务所组成的软件部分。其中，机器人可采用上海广茂达公司的 AS-MRobotE 模块机器人，它们通过 100Mb/s 的局域网通信。应用服务层所包含的服务有：用户交互服务、路径规划服务和运动控制服务。用户交互服务为系统使用者提供人机接口，负责与系统使用者交互，提供友好的人机界面，系统使用者能够方便地对控制平台按照自己的需求进行系统配置，并且对系统发送各种级别的命令控制，同时，系统将执行状态以及各种数据反馈给系统使用者。路径规划服务是智能处理的中心，它包括一系列的算法接口以及智能分析接口，使用这些算法产生具体命令；作为最高层的智能模块对机器人的行动起指导性作用。运动控制服务管理机器人的底层运动规划，主要功能包括机器人的轨迹规划以及运动插补、反解运算等；主要对机器人的速度、加速度以及位移进行控制，并且将产生的运动信息按照命令格式，转换成机器人命令进行存储或者发送。

本发明基于LSOA架构的开放式控制方法，其基本控制流程如图4所示，包括如下步骤：

S1、应用服务层向总线控制层提交注册服务的请求。一个应用服务要成为系统中的一个节点，接受总线的管理，和系统中的其它节点进行交互，就必须先在总线上进行注册。需要提交的注册信息包括服务的名称，服务所提供的方法及其原型描述。注册成功后应用服务会得到一个由总线分配的服务 ID。

S2、用户向总线控制层配置服务流程。由于本系统采用 SOA 架构，松耦合的组合方式使各应用组件能保持独立性，很容易被重用或重新创建而互不干扰，因此我们可以通过增加或删除服务，在总线控制层（也称服务总线）上组织新的业务逻辑来实现新的应用。对于不同领域的应用，其设备层、应用服务层等均具有特殊性，所以业务流程必然具有差别；本发明中，由总线控制层实现对数据流的逻辑管理，允许客户在总线控制层的控制界面中通过列表选择应用服务的方法来设置应用逻辑的业务流程，添加新的应用逻辑，这样增加了系统的灵活性及可扩展能力。定制控制系统的应用组件的过程可以看作是配置服务的过程。服务流程的配置过程如图 所示，主要包括以下几个步骤：

S21、流程配置程序从总线内核的数据库获取所有服务的相关信息列表。

S22、选取一个服务当中提供的某个方法功能，将其作为新的流程节点添加到服务流程列表中。

S23、流程配置程序将根据步骤 S22 中用户所选择的流程节点的输出参数类型，更新可添加的服务方法信息列表。这样做的目的是使得服务流程中前一级服务的输出参数类型和数量必须与后一级服务的输出参数类型和数量一致。

S24、如果需要添加新的服务，则返回步骤 S22 继续添加；否则完成服务流程配置。

S25、为刚配置好的服务流程生成服务流程 ID。这个 ID 在同一个总线内核中是唯一的。

S26、步骤 S25 的操作成功后，则将配置好的服务流程信息保存在总线内核的数据库中。

S3、用户向总线控制层发送流程请求消息。用户要调用某一流程完成特定的控制逻辑时，需要调用传输层提供的 API 向总线控制层发送流程请求消息。请求信息需要包含所请求的流程 ID、相应的请求参数。

本发明所基于的控制系统采用一种二进制的消息协议；系统中各个进程间传递的消息必须遵循此消息协议。相比起 Web Service 所采用的基于 XML 的 SOAP 消息协议，本发明所基于的系统采用的协议由于免去了文本解析等开销，效率上会有很大的优势；其消息传输分层架构如图 所示，其中层次具体功能如下：

应用层：包括应用服务、设备服务和总线内核。

消息传输底层库：作为消息传输分层架构中的最底层，消息传输底层库为上层提供最基本的功能：将消息准确无误并且高效的从发送方传输到接收方。值得注意的是，消息传输底层库

并不对消息中包含的数据进行解析，它只负责将消息可靠无误的传输到接收方，至于消息应该如何封装，里边应该包含什么数据，这是上层的功能。

为了更高效的让系统中的进程处理消息的发送和接收，本发明采用了信号量的机制去同步消息发送方和接收方的行为。这使得当消息发送方发现目标共享内存已满时，将会进入休眠状态，一旦接收方从共享内存中取走消息后，将会通知休眠中的消息发送方继续发送消息。同理，当消息接收方发现当前没有可以处理的消息时，将会进入休眠状态，直到消息发送方发送了新的消息后才会被唤醒。这种机制有效提高了收发消息进程的效率，避免了忙等待或固定间隔轮询所造成的低性能表现。

传输支持层：传输支持层构建在消息传输底层库之上，为上层的总线内核以及服务提供消息传输的接口，其原理如图 所示。传输支持层分为两部分，一部分是面向应用服务以及设备服务的**服务支持 API**；另一部分是面向总线内核的**总线支持 API**。**服务支持 API**与**总线支持 API**都被设计成是基于连接的通讯方式，而将底层实现的基于共享内存的模式隐藏起来，有利于简化程序结构，将程序开发者注意力集中在业务逻辑上。传输支持层支持消息通知机制。和普通的消息不同，发送通知消息并不需要指定流程逻辑 ID，只要发送到总线上，总线会根据配置转发到订阅了这个通知的服务上。

**S4、总线控制层根据流程请求消息的优先级对流程请求进行调度。**

本发明的调度模块采用不可抢占式的最早截止时间优先（EDF）调度算法。每个流程请求的任务被分为多个子任务，其中每个服务被看做一个子任务，子任务执行不允许被打断。可调度性分析根据本发明特有的算法对子任务集进行可调度性认证。只有通过可调度性认证的子任务集才能在系统中运行。调度器通过修改每个服务的优先级来实现对子任务的调度。

当一个流程请求到达时，调度器会通过该流程请求的周期，各个子任务的最大执行时间，截止时间等先进行可调度性认证。将第一个子任务加入到就绪任务集中。在每个子任务执行结束时根据就绪子任务的最早截止时间选择子任务执行。

**S5、总线控制层对流程请求进行处理，然后向相应服务发送流程请求。**

服务请求端发送到总线控制层的信息类型包括：服务请求、状态信息、总线同步信息包和注册信息包。总线控制层监听到不同信息类型的数据包后执行相应的处理流程，这些流程包括：请求信息处理流程、注册信息处理流程、状态信息处理流程（即容错管理流程）和总线同步信息处理流程等。

为达到以上功能需求，总线控制层（本发明中，也称为软总线）应具有如图 所示的功能模块：

信息中心。软总线在能正常处理消息之前，必须先对软总线的信息中心进行配置，其中包括服务列表、请求列表、任务列表。服务列表主要包含服务的注册信息，如服务 ID、服务地

址、服务消息协议信息、服务状态信息等。请求列表包含每个请求的信息如请求的优先级、请求所需调用的任务等。任务列表包含任务的业务逻辑，即需要调用哪些服务及服务调用的顺序。

消息处理模块。是总线控制层功能的主要实现模块，解析消息头获取消息属性并进行相应操作，比如将其发送到对应的目标组件或更新目录。

调度模块。采用调度器软件统一调度管理各任务构件，控制各个任务构件无冲突地分时访问软件总线。

安全管理模块。安全管理包括服务权限管理和数据的加密解密管理。

容错模块。（1）当连接代理模块检测到在线状态的任务列表里的服务不在线，则对该服务发送控制包，进行服务端与总线的重连处理。（2）当连接代理模块反馈信息显示两服务端连接失败时，发送控制包，重新建立两服务端的连接。

业务流程配置模块。通过友好的人机界面配置服务调用的业务流程，而且可实现在系统运行时动态配置系统的运行流程。

当总线接收到注册（或注销）信息，则修改目录，增加（或删除）该服务信息。注册信息的处理流程如图 所示，包括以下步骤：

S511、总线内核接收到请求消息；

S512、总线内核对请求消息进行分析，判断是否注册请求消息，是则到步骤 S513，否则执行其它请求处理流程；

S513、总线内核开启线程处理注册请求；

S514、总线内核为服务分配服务 ID，这个 ID 是本总线内核实例范围中唯一的；

S515、总线内核将服务信息（包括服务 ID、服务名称、服务方法等）写入总线内核数据库，并将服务 ID 发送给请求注册的服务，使得服务下次启动的时候无需再次请求注册，只需要以总线分配的服务 ID 登记启动即可。

总线监听各服务端消息的流程如

图所示，各类信息存入队列时按照优先级顺序插入到队列中，执行时从队列头取出第一条信息开始执行。总线对各队列进行监听，当有消息时则进行相应处理。具体包括以下步骤：

S521、总线监听消息队列，当存在消息需要处理时，转到 S522；

S522、总线内核对消息进行分析，检查消息请求源端是否合法，是则转到步骤 S523，否则舍弃此请求消息；

S523、对请求消息进行分析，判断是否流程请求消息，是则将其存入流程请求列表，否则转步骤 S524；

S524、判断是否服务注册信息，是则将其存入注册信息队列，否则转步骤 S525；

S525、判断是否服务状态信息，是则将其存入状态信息队列，否则转步骤 S526；



S526、将当前请求信息存入其它信息队列。

请求信息处理流程如图 2所示，调度模块应监控队列中各请求的执行时间，当时限超过时丢弃改请求，返回超时信息或把该请求信息调到队列的最末端，即降低该请求信息的优先级。具体包括以下步骤：

S531、监听流程请求队列，当存在请求信息待处理时，转步骤 S532；

S532、查找请求所需调用的服务流程，若存在查找成功则转 S533，否则向服务请求端发送调用服务失败信息，从消息队列中删除该请求；

S533、按预先定义好的流程设置执行总线提供的流程，调用相应的服务方法；

S534、请求处理完毕请求信息后，从请求队列中删除该请求信息，重新进入等待状态。

总线接收到状态信息后首先更新连接代理模块中的服务端在线状态和服务之间连接状态的信息，当出现异常时，做出异常处理，若有后备模块则查找后备模块并替换，并通知服务（客户）端定位到后备服务，否则通知各个服务（客户）端停止计算。状态信息处理流程（即容错管理流程）如图 3所示。具体包括以下步骤：

S541、总线内核监听状态信息队列，当存在状态信息待处理时转步骤 S542；

S542、分析状态信息，将内含的服务状态信息和服务连接状态信息更新到总线内核维护的状态信息中；

S543、判断服务状态是否出现异常，是则转步骤 S544，否则转步骤 S546；

S544、检查是否有替换服务。替换服务是在流程定义的时候设置的，每一个流程节点都可以设置一个或多个的替换服务方法。若存在替换服务则转步骤 S545，若不存在替换服务则通知系统中的各个服务停止运行与该出现异常的服务相关的程序；

S545、将替换了的服务信息在总线数据库中进行更新。

S6、应用服务收到请求消息，执行完指定的控制逻辑后向总线控制层返回结果。

应用服务接收到请求消息后，将对请求消息进行解析，提取出目标方法以及调用参数，传递给服务内部的方法执行。当指定的服务方法执行完毕后，结果将通过服务支持层提供的 API 返回给总线控制层。

S7、总线控制层将结果返回给用户。

总线控制层接收到来自应用服务的结果后，将根据流程配置的信息将结果返回给相应的用户，所述结果可能是调用流程服务的结果，也可能是程序序列中的下一个服务的结果。

上述实施例为本发明较佳的实施方式，但本发明的实施方式并不受上述实施例的限制，其他的任何未背离本发明的精神实质与原理下所作的改变、修饰、替代、组合、简化，均应为等效的置换方式，都包含在本发明的保护范围之内。

# 权 利 要 求 书

---

1、基于LSOA构架的开放式控制方法，其特征在于，包括以下步骤：

- S1、应用服务层向总线控制层提交注册服务的请求；
- S2、向总线控制层配置服务流程；
- S3、向总线控制层发送流程请求消息；
- S4、总线控制层根据流程请求消息的优先级对流程请求进行调度；
- S5、总线控制层对流程请求进行处理，然后向相应服务发送流程请求；
- S6、应用服务收到请求消息，执行完指定的控制逻辑后向总线控制层返回结果；
- S7、总线控制层将结果返回。

2、根据权利要求 1 所述的基于 LSOA 构架的开放式控制方法，其特征在于，步骤 S2 所述配置服务流程包括以下步骤：

- S21、流程配置程序从总线内核的数据库获取所有服务的相关信息列表；
- S22、选取一个服务当中提供的某个功能，将其作为新的流程节点添加到服务流程列表中；
- S23、流程配置程序将根据步骤 S22 中所选取的流程节点的输出参数类型，更新可添加的服务流程信息列表；
- S24、如果需要添加新的服务，则返回步骤 S22 继续添加；否则完成服务流程配置；
- S25、为刚配置好的服务流程生成服务流程 ID；
- S26、步骤S25的操作成功后，则将配置好的服务流程信息保存在总线内核的数据库中。

3、根据权利要求1所述的基于LSOA构架的开放式控制方法，其特征在于，步骤S5所述流程为：请求信息处理流程、注册信息处理流程、状态信息处理流程或总线同步信息处理流程。

4、根据权利要求 3 所述的基于 LSOA 构架的开放式控制方法，其特征在于，当所述流程为注册信息处理流程时，注册信息处理流程包括以下步骤：

- S511、总线内核接收到请求消息；
- S512、总线内核对请求消息进行分析，判断是否注册请求消息，是则到步骤 S513，否则执行其它请求处理流程；
- S513、总线内核开启线程处理注册请求；
- S514、总线内核为服务分配服务 ID，这个 ID 是本总线内核实例范围中唯一的；
- S515、总线内核将服务信息写入总线内核数据库，并将服务ID发送给请求注册的服务。

5、根据权利要求 3 所述的基于 LSOA 构架的开放式控制方法，其特征在于，当所述流程为请求信息处理流程时，请求信息处理流程包括以下步骤：

- S531、监听流程请求队列，当存在请求信息待处理时，转步骤 S532；

S532、查找请求所需调用的服务流程，若查找成功则转 S533，否则向服务请求端发送调用服务失败信息，从消息队列中删除该请求；

S533、按总线提供的流程，调用相应的服务；

S534、处理完请求信息后，从请求队列中删除该请求信息，重新进入等待状态。

6、根据权利要求 3 所述的基于 LSOA 构架的开放式控制方法，其特征在于，当所述流程为状态信息处理流程时，状态信息处理流程包括以下步骤：

S541、总线内核监听状态信息队列，当存在状态信息待处理时转步骤 S542；

S542、分析状态信息，将内含的服务状态信息和服务连接状态信息更新到总线内核维护的状态信息中；

S543、判断服务状态是否出现异常，是则转步骤 S544，否则转步骤 S546；

S544、检查是否有替换服务。替换服务是在流程定义的时候设置的，每一个流程节点都可以设置一个或多个的替换服务方法。若存在替换服务则转步骤 S545，若不存在替换服务则通知系统中的各个服务停止运行与该出现异常的服务相关的程序；

S545、将替换了的服务信息在总线数据库中进行更新。

# 说明书摘要

---

本发明基于LSOA构架的开放式控制方法，包括以下步骤：应用服务层向总线控制层提交注册服务的请求；向总线控制层配置服务流程；向总线控制层发送流程请求消息；总线控制层根据流程请求消息的优先级对流程请求进行调度；总线控制层对流程请求进行处理，然后向相应服务发送流程请求；应用服务收到请求消息，执行完指定的控制逻辑后向总线控制层返回结果；总线控制层将结果返回。本发明是一个具有开放式系统结构、可扩展、可重构、可在线升级维护、应用服务容错能力强等特点的控制方法。

# 说明书附图

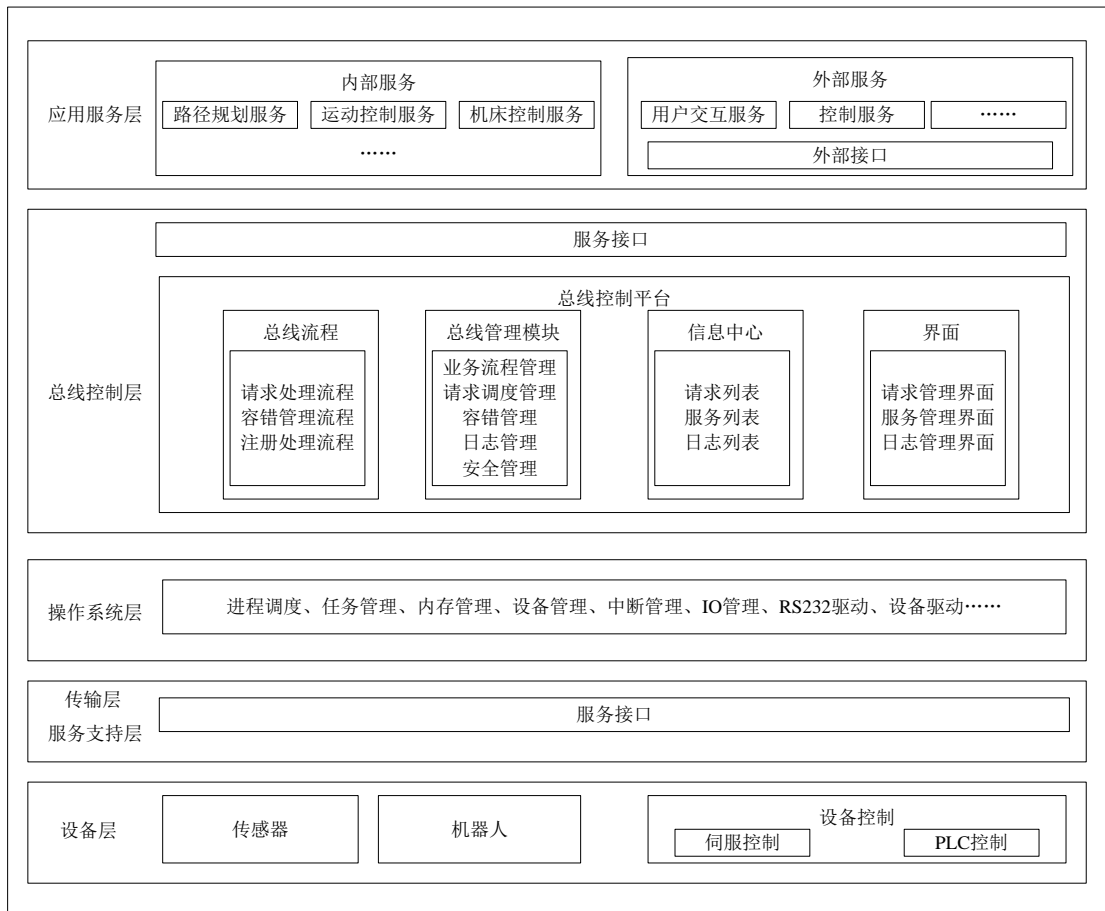


图 1

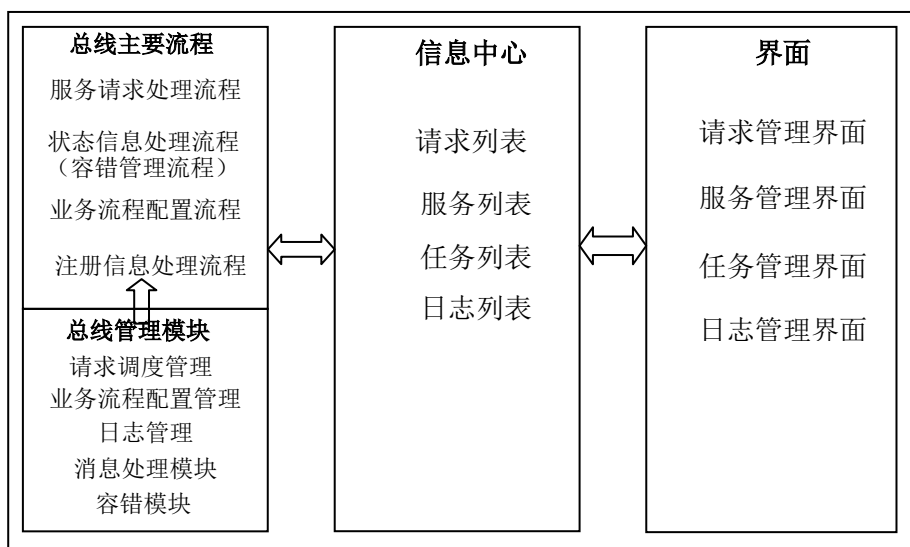


图 2

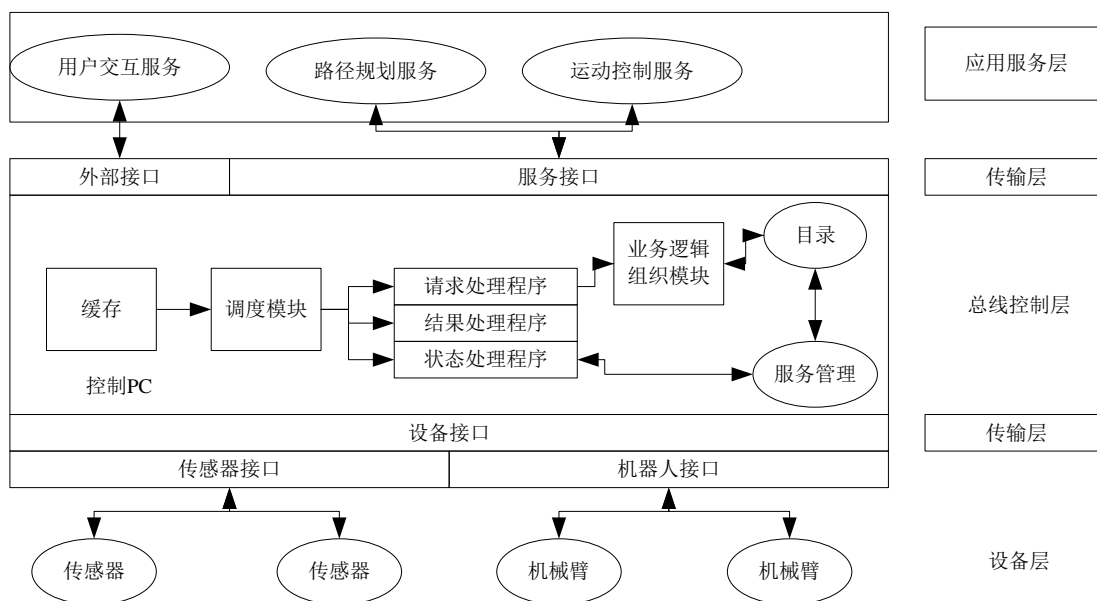


图 3

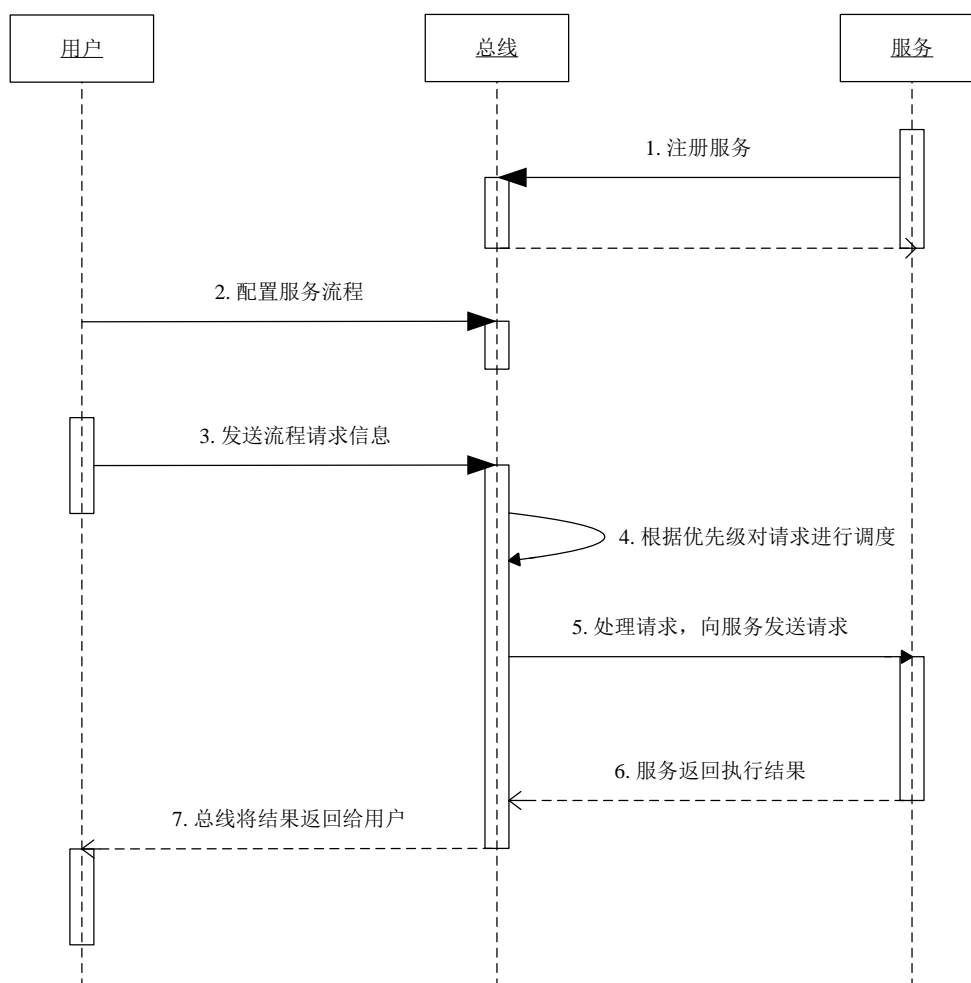


图 4

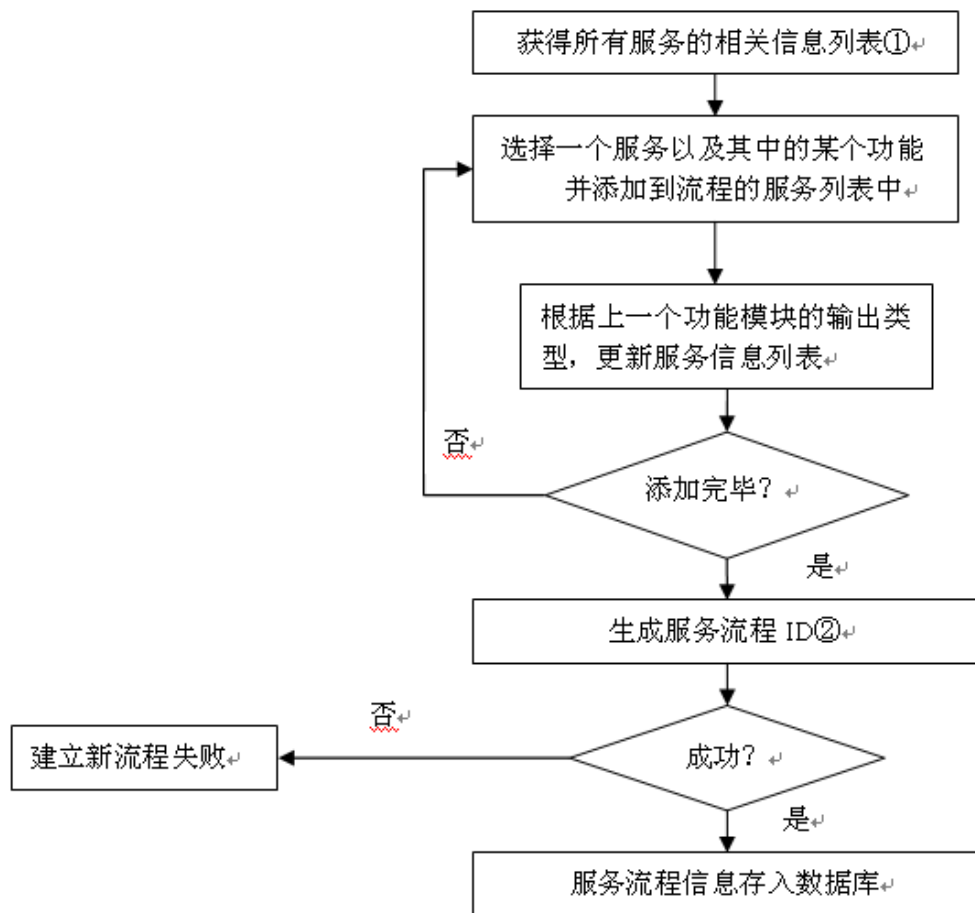


图 5

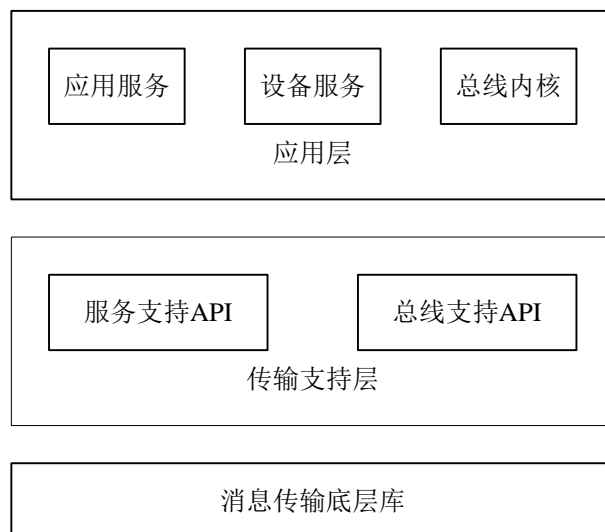


图 6

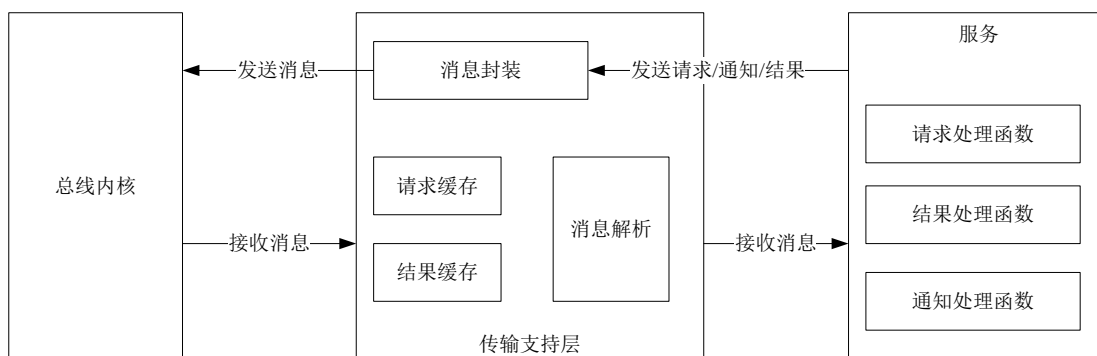


图 7

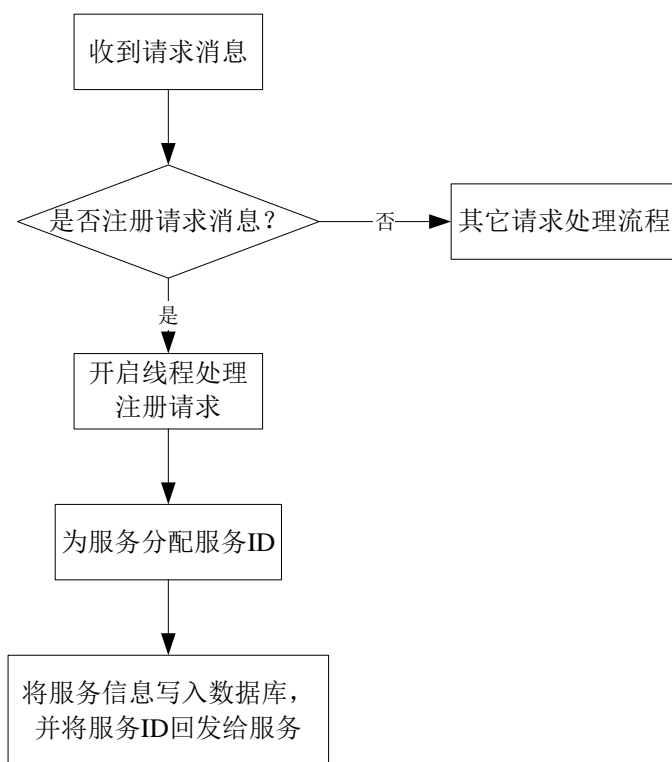


图 8

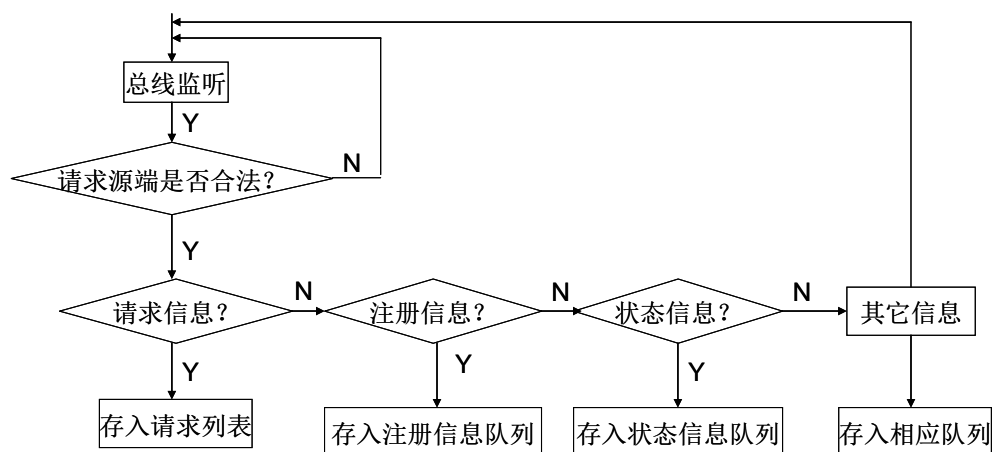


图 9



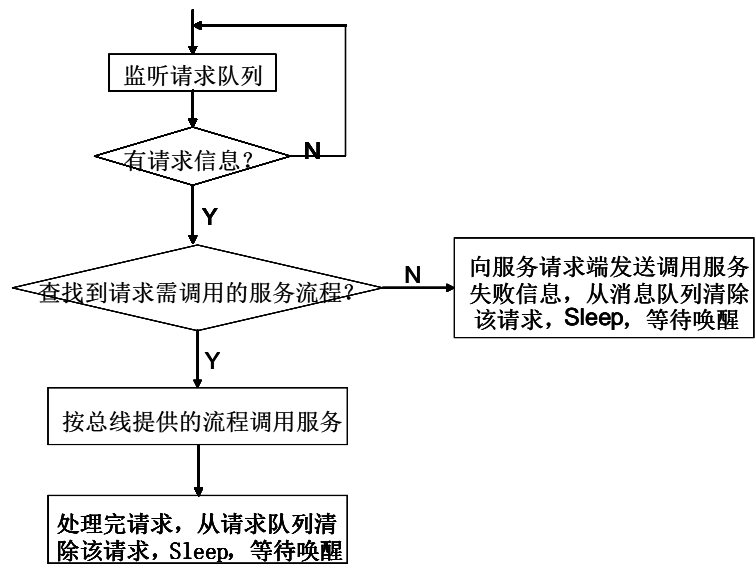


图 2

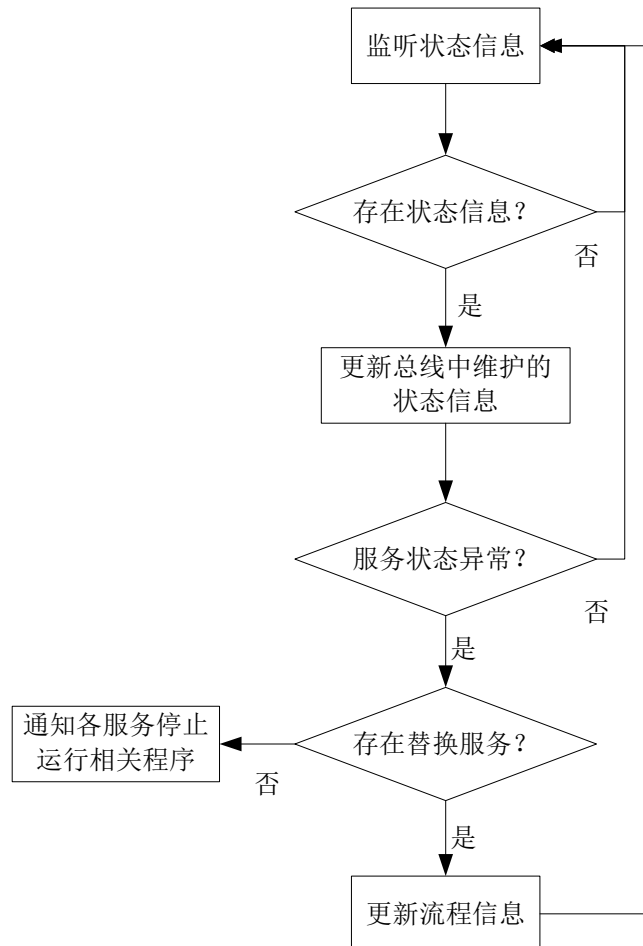


图 3

# 摘要附图

---

