

SUNDAYMAKER TEAM

概要设计

SundayMaker Basic Layer(SMBL)-基于 ASP.net2.0 的网站通用底层

陈魏明

4/6/2009

该文档概述了 SMBL 的架构和设计，明确需求规约中提出的对网站安全性、可扩展性及性能方面的要求。版权所有 2009

版本控制		
版本号	作者	目的
1.0	陈魏明	提出底层架构整体设计。

目录

1. 引言	5
1.1. 编写目的	5
1.2. 背景	5
1.3. 定义	5
1.4. 参考资料	5
2. 总体设计	6
2.1. 需求规定	6
2.2. 运行环境	6
2.2.1. 硬件环境	6
2.2.2. 软件环境	6
2.3. 基本设计概念和处理流程	6
2.3.1. 基本设计概念	6
2.3.2. 处理流程	7
2.4. 结构	9
2.4.1. 总体架构	9
2.4.2. 启动引导区/全局数据子系统	9
2.4.3. 内核级服务子系统	10
2.4.4. 总控子系统	10
2.4.5. 系统抽象	11
2.4.6. 操作抽象	11
3. 接口设计	11
3.1. 外部接口	11
3.2. 内部接口	11
4. 运行设计	12
4.1. 运行模块组合	12
4.2. 运行控制	14

4.3.	运行时间	14
5.	系统数据结构设计	14
5.1.	逻辑结构设计要点	14
5.2.	物理结构设计要点	15
5.3.	数据结构与程序的关系	15
6.	系统出错处理设计	15
6.1.	出错信息	15
6.2.	补救措施	15

1. 引言

1.1. 编写目的

该文档确定网站底层整体架构方案，明确各模块之间的工作流程。将通用底层架构设计方案记录在册，便于测试工作展开及日后维护工作。

1.2. 背景

- a. 项目名称：SundayMaker Basic Layer(SMBL)-基于 ASP.NET2.0 的网站通用底层
- b. 任务提出：陈魏明 (littlepush@gmail.com)、孙玮 (sunwei2321@gmail.com)
- c. 开发者：陈魏明、孙玮

1.3. 定义

SM: SundayMaker , 项目内部代号

BL: Basic Layer , 基本层 , 底层

ASP.NET 2.0: 由 Microsoft 公司开发的 Web2.0 项目开发框架

ADBS: Abstract Database System , 抽象数据库系统

AFS: Abstract File System , 抽象文件系统

1.4. 参考资料

《SMBL 需求规约》

《概要设计说明书 (GB8567-88)》

2. 总体设计

2.1. 需求规定

具体需求规定请参阅本项目的需求规约文档，此处不再说明和举例。

2.2. 运行环境

2.2.1. 硬件环境

- PC 环境
- 处理器: P4 2.0GHz 及以上
- 内存：1GB 及以上
- 自适应网络适配器支持 10/1000 M/bs
- 硬盘 80G 及以上

2.2.2. 软件环境

- 操作系统：Windows Server 2003
- 服务要求：IIS6.0，ASP.NET 2.0
- Ajax Extension 1.0

2.3. 基本设计概念和处理流程

2.3.1. 基本设计概念

本系统作为一个网站的通用底层，按照传统的网站三层架构进行划分，可将其视为最底部的数据抽象层(DAL)。本系统同时面向中间的业务逻辑层 (BLL) 提供了统一的 API。

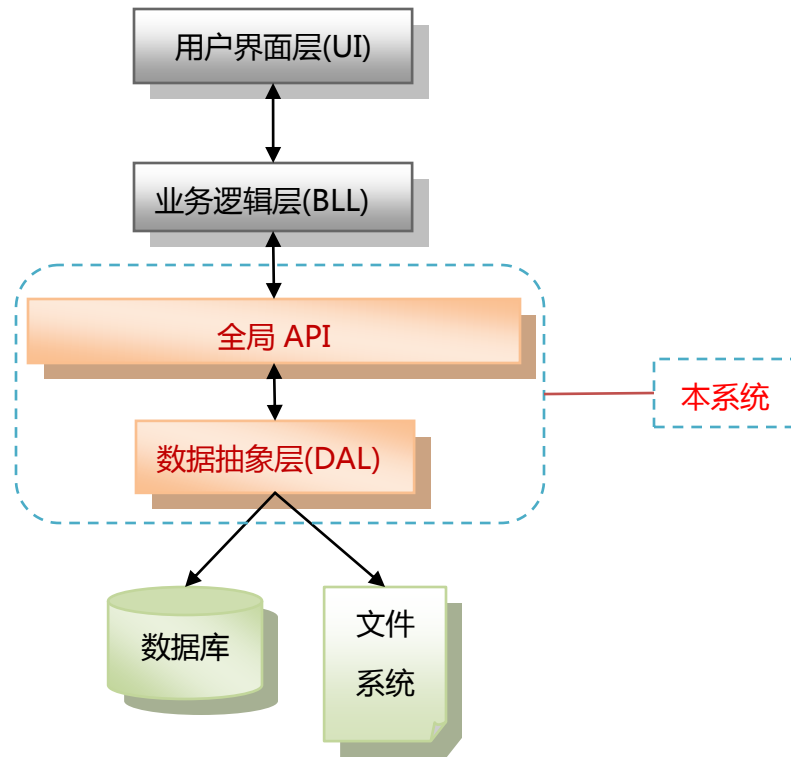


图 2-1 系统在网站应用中的层次结构

该系统在满足基本数据抽象层的需求之余，提供了全局的错误管理机制，将所有的异常进行统一处理。

2.3.2. 处理流程

➤ 启动时

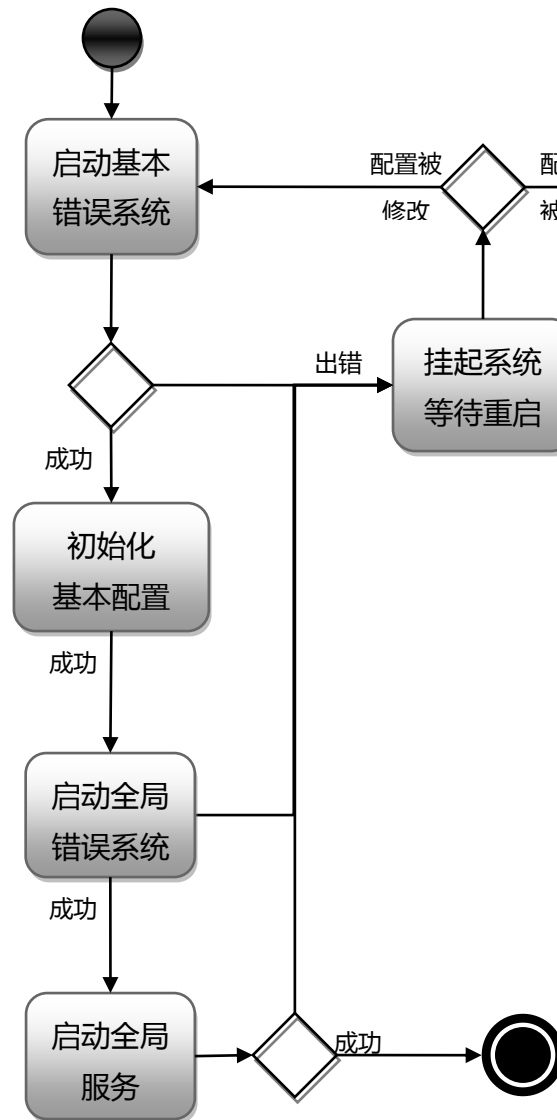


图 2-2 系统启动时流程

➤ 运行时

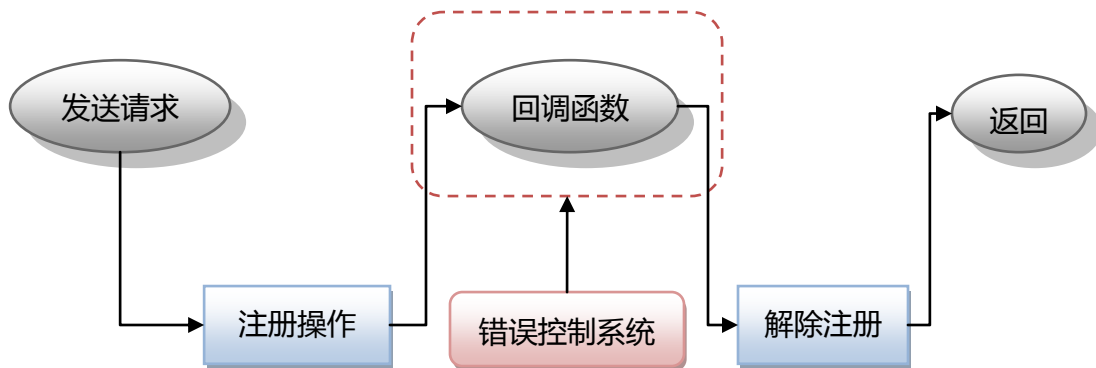


图 2-3 系统运行时流程

2.4. 结构

2.4.1. 总体架构

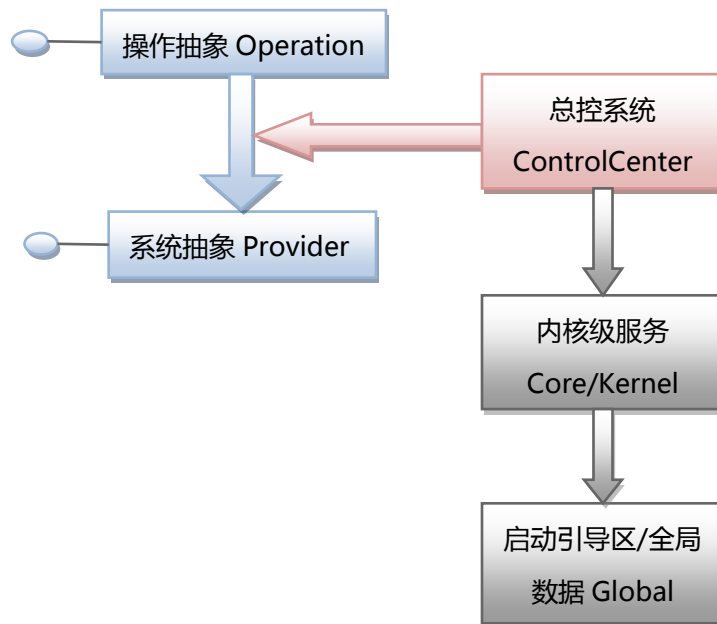


图 2-4 总体架构图

2.4.2. 启动引导区/全局数据子系统

该子系统为整个系统的启动进行引导，作为第一个启动的项目，负责初始化基本错误系统，载入配置文件，初始化全局数据的工作。

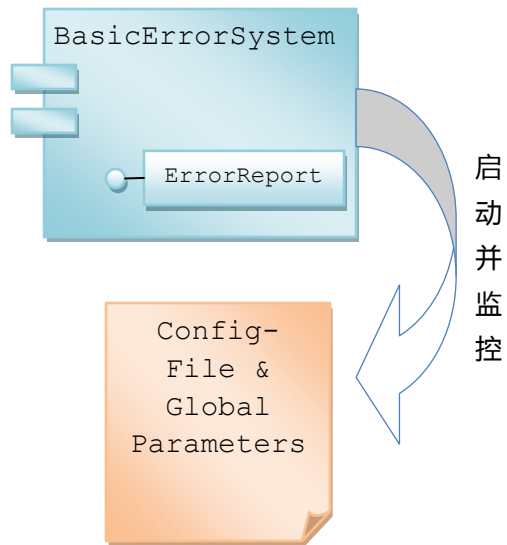


图 2-5 启动引导区/全局数据

2.4.3. 内核级服务子系统

该子系统为整个系统的核心服务，负责对配置文件的管理，包括解析和修改。该系统启动后同时启动错误系统和日志系统。

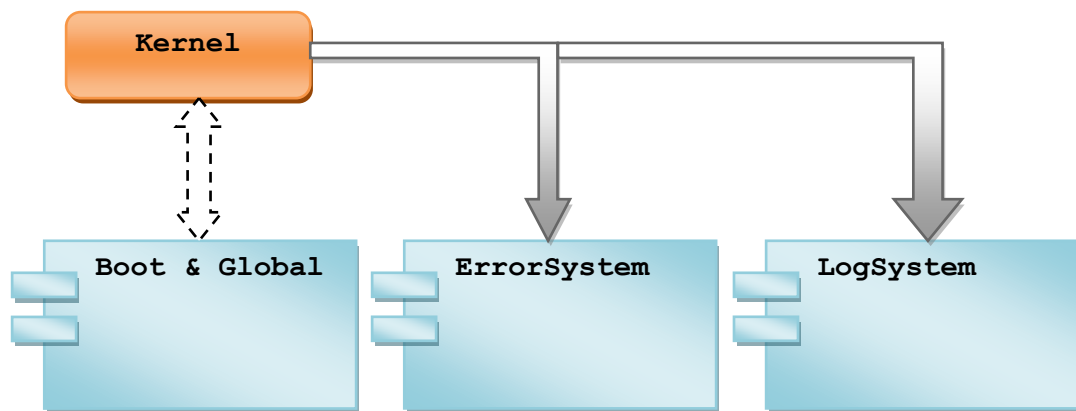


图 2-6 内核与引导/全局系统的关系

2.4.4. 总控子系统

该子系统为整个系统面向上层应用的调度和管理接口。所有的业务逻辑层的操作都由该子系统负责统一管理，所有的底层资源由该子系统负责调度。

2.4.5. 系统抽象

该部分可分为两个小型子系统：抽象数据库系统和抽象文件系统。这两个系统分别负责对数据库连接和文件访问的管理，并提供统一的服务接口。

2.4.6. 操作抽象

该部分是对所有操作的抽象封装。是整个系统暴露给上层业务逻辑的直接接口。任何的操作都需要从该接口继承才能时候本系统提供的各项服务。

3. 接口设计

3.1. 外部接口

网络协议：TCP/IP, HTTP

文件访问接口：System.IO

数据库默认提供接口：ODBC 驱动，ADO.NET 驱动

3.2. 内部接口

总控系统(ControlCenter)、日志系统(LogSystem)与错误系统(ErrorSystem)的接口：异常类(Exception Class)，错误信息(Error Message)

总控系统和日志系统作为作为错误系统的上层调用，与错误系统之间通过 Exception Class 的实例或明确的 Error Message 传递信息。

在总控系统捕获异常或某个操作返回失败时，通过将该异常直接传递给错误系统或者将一条经过编排的错误信息传递给错误系统。

错误系统(ErrorSystem)与基本错误系统的接口：异常类(Exception Class)，错误列表(Error List)

错误系统和基本错误系统两个子系统之间的信息传递通过错误系统在运行过程中捕获的异常和期间积累的错误记录发生。

错误系统在运行过程中发生未能正常处理的问题，或者无法获得足够的文件系统权限时，会抛出异常信息，这些异常将交由基本错误系统处理。当错误系统无法处理错误池中的错误列表时，也将交由基本错误系统处理。

内核(Kernel)与基本错误系统的接口：异常类(Exception Class)

内核在执行过程中发生不可处理异常时，通过将该异常传递给基本错误系统进行处理。

操作抽象(Operation)与系统抽象的接口：供应商接口(Provider)

操作抽象中的任何一个操作类，均通过总控系统提供的供应商接口与系统抽象进行交互。供应商接口提供与系统资源有关的所有操作。

4. 运行设计

4.1. 运行模块组合

启动并常驻内存的模块：

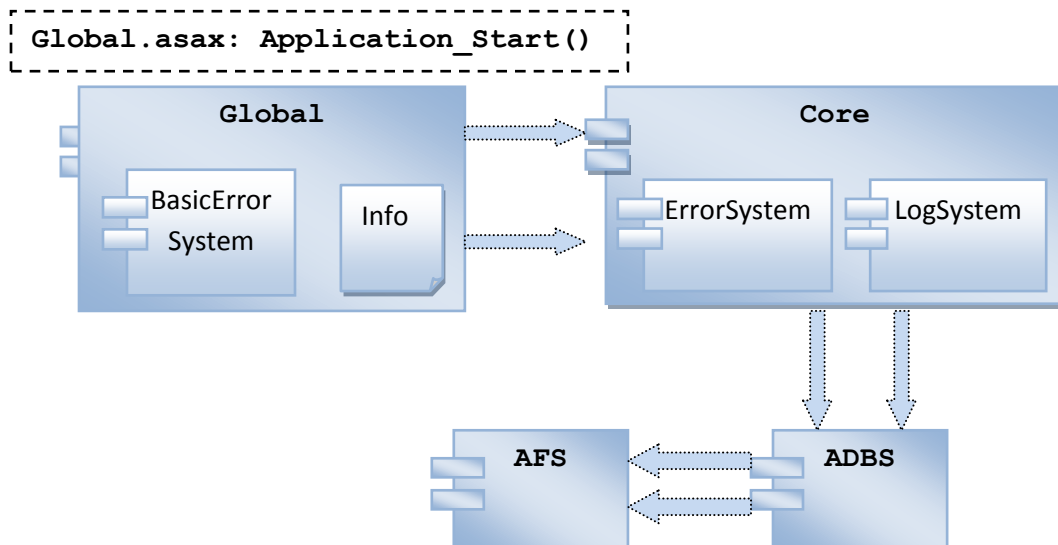


图 4-1 启动并常驻内存模块组合

Application_Start 为 ASP.NET 提供的回调函数，负责控制本系统的启动顺序。

数据库操作：

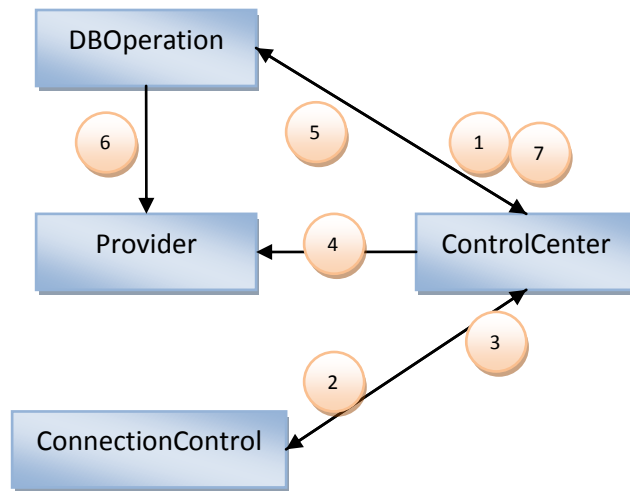


图 4-2 数据库操作模块组合

1. Operation 向 ControlCenter 注册
2. ControlCenter 向 ConnectionControl 申请可用数据库连接
3. ConnectionControl 分配连接
4. ControlCenter 启动/初始化连接
5. ControlCenter 完成注册，并返回连接给 Operation
6. Operation 通过转换成 Processor 的 Provider 执行操作
7. Operation 完成操作后向 ControlCenter 申请解除注册并释放资源

文件操作：

文件操作的流程和数据库操作相同，其中数据库的 ConnectionControl 对应为文件系统中的 DocumentPool，其余模块不变。

4.2. 运行控制

仅有一种运行模式，且有系统按顺序调用

4.3. 运行时间

启动：由配置文件大小决定载入文件所需的时间

内核初始化：由配置文件中的内容数量决定读取文件的时间

数据库系统：上限 30 秒

文件系统：由文件大小及操作数决定

5. 系统数据结构设计

5.1. 逻辑结构设计要点

本系统中的数据集主要为配置文件各元素对应的数据记录。

类名	字段	类型	说明
DBProviderInfo	Name	String	Provider 的标示
	TYPE	ProviderType	Provider 的类型 (SQLDB/OLEDB/ORALCE/MYSQL)
	MaxConn	Int32	最大连接数
	DBName	String	数据库名称
	DBPath	String	数据库文件名或服务器地址
	UserName	String	访问数据库的用户名
	Password	String	访问数据库的密码
ConnectionString	TYPE	ProviderType	Provider 的类型 (SQLDB/OLEDB/ORALCE/MYSQL)
	InnerText	String	ConnectionString 所包含的数据库连接字段
FileLocation	Title	String	文件标题/文件名
	Path	String	文件的相对路径
	MaxSize	Int32	单个文件的最大值

	MaxCacheSize	Int32	单个文件写入缓存区大小
PageLocate	Code	String	页面编号
	Path	String	页面相对路径
	Description	String	页面描述

5.2. 物理结构设计要点

本系统的数据在运行时保存在内存中，同时在部署的网站中有独立的文件保存。所有信息只能通过具有管理员权限的帐号经由特殊管道修改。

5.3. 数据结构与程序的关系

配置文件仅在系统启动时载入，并经处理后保存在内存中，运行时不访问配置文件以及配置文件对应的数据结构。

6. 系统出错处理设计

6.1. 出错信息

- 配置文件丢失
- 没有写入权限
- 网络连接中断（仅 localhost 可见）

6.2. 补救措施

所有的基本错误均由 BasicErrorSystem 处理，处理方式为发送特定格式的错误报告至 smb1.error@gmail.com，同时上层网站如果对错误进行处理，则按照网站罗基层进行处理，如果没有处理，则跳转到系统错误页面，提示网站出现严重错误。