GS-9330-Sender

**软件架构设计文档**

**成都驰通数码系统有限公司**

# 修改记录

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 作者 | 版本描述 | 日期 |
| 1.0 | 刘浩 | 第1版 | 2015/12/7 |
|  |  |  |  |

# 目录

修改记录 1

目录 i

1 概述 1

1.1 目的 1

1.2 对象与范围 1

1.3 名词与术语 1

1.4 文档的组织结构 1

2 总体结构的分析与设计 2

2.1 设计目标与原则 2

2.2 设计策略一：xxxxxx 2

2.3 设计策略二：xxxxxx 2

3 总体功能的分析与设计 3

3.1 设计目标与原则 3

3.2 设计策略一：xxxxxx 3

3.3 设计策略二：xxxxxx 3

4 软件模块说明 4

4.1 模块关系图 4

4.2 模块一：xxxxxx 4

4.3 模块二：xxxxxx 4

附录1 参考文献 5

修改记录 6

# 概述

说明：在此部分分节说明编写此文档的目的和主要内容；指明此文档的读者对象和生效范围；最后对此文档所使用的专用术语，规范以及文档的组织结构分别加以介绍。

## 目的

## 对象与范围

本文档的主要目的是介绍Sender的系统构架。 包含构架相关的策略，以及和这些策略相关的功能/性能/可用性/可扩充性相关的需求。本文档的主要读者对象为系统的设计人员，开发人员，测试人员（白盒测试）。

## 名词与术语

| 缩写、术语 | 解释 |
| --- | --- |
| DVB | 数字视频广播 |
|  |  |
| … |  |

## 文档的组织结构

# 总体结构的分析与设计

*说明：本章通过对影响和制约软件结构的各种需求与约束的分析，为总体结构的设计制定相应的策略。本章主要关注的焦点是：（1）软件的模块如何划分（2）模块之间的相互关系与通信机制。与总体结构无关的设计策略，应在第三章中描述，不包含在本章的范围之内。*

## 设计目标与原则

*说明：本小节明确总体结构的设计目标，列出设计时所必须遵循的大原则，以及遵循或参考的标准，如J2EE, TMF等等。*

Sender系统有以下几个非功能性需求需要在系统的构架中体现：

1. 可移植性， 系统需要运行于Linux 和 Windows 平台。
2. 高可用性， 系统需要24\*7不间断提供服务。
3. 可扩充性， 将来可用需要增加的功能有： 新的table，如TOT。 新的Descriptor。增加用户界面以方便用户配置发送相关的参数。暂停/继续某一张表的发送。

所以，我们的构架必需确保我们将来可以方便安全的添加这些功能且新功能不会对已有功能产生负面影响。

## 设计策略一：3层体系结构

在一个严格分层的结构中, 第n层仅仅使用n-1层或更低层所提供的服务。 本软件系统使用 《Patterns of Enterprise Application Architecture》第1.2 章节推荐的3层结构划分，如下图所示：



我们的构架将整个软件分为3个层次：Presentation Logic Layer， Business Logic Layer，Data Access Layer。在实现设计过程中，以上提到的3个层次可能会被进一步划分为更多的层次，以便减少模块间的耦合，提高软件系统的可移植性。

1. Presentation Logic Layer: 当前，Sender应用程序暂时不会提供可供用户配置发送过程/监控发送状态的用户界面。 但是我们的确在需求分析过程中发现了一些可供用户配置的软件元素，如：发送时间间隔配置，暂停/继续发送。 我们在构架设计阶段就按照3层结构来设计Business Logic Layer， 以便将来可以方便安全的添加用户界面。
2. Business Logic Layer: 这一层主要负责所有和业务逻辑相关的工作， 如: 根据网络上下级关系计算需要发送的Transform Packet 的内容， 根据发送间隔的配置调度Transform Packet发送。
3. Data Access Layer: 本软件系统可支持2种永久数据的读写，即XML 和 TS文件， 将来很可能会增加新的数据读写方式，如MySql，SqlServer数据库。 DAL的功能就是读取永久数据，创建业务对象，并通过调用回调函数将数据的变化通知到上层。 业务对象的定义属于Business Logic Layer的内容， 为了避免下层依赖对上层的依赖，本层所有的类都是C++模板内，其模板参数的类型就是可能的业务对象的类型， Business Logic Layer通过模板参数类型的对象修改业务对象的属性。此设计符合OOD的”依赖倒置原则”。

## 设计策略二：构架描述标准

本构架描述文档按照ISO/IEC 42010:2011 (“Architecture description”)的要求来描述Sender软件系统的构架设计, 标准命名中的2011 表示此标准的发布时间为2011年。

ISO/IEC/IEEE 42010:2011 defines requirements on the description of system, software and enterprise architectures. It aims to standardize the practice of architecture description by defining standard terms, presenting a conceptual foundation for expressing, communicating and reviewing architectures and specifying requirements that apply to architecture descriptions, architecture frameworks and architecture description languages.

## 设计策略三：可移植性策略

Sender 需要运行于Windows 和 Linux平台。 为了使同一套代码运行于多个OS平台， 我们使用ACE C++库来隔离我们的代码对OS提供的API的依赖。

另外，我们采用C++11作为我们的编程语言， 以提高我们系统的可移植性。 如：创建线程时，我们不调用Windows和Linux提供的创建线程的API，而是直接调用C++提供的线程函数。

## 设计策略四：事件/Timer/多线程策略

Sender 在运行的过程中，需要处理多种事件: Timer时间， 添加/删除XML文件事件， 等等。 我们需要处理的事件将会随着需求的增加而增加， 如：新的需求可能会要求我们动态的处理网络地址变化的事件。

为了系统的可扩充性和可移植性，我们采用ACE Reactor Framework作为我们的应用程序的主要框架。 并使用ACE提供的Timer, Active Object… 等基础设施来完成timer, 多线程相关的实现。

# 总体功能的分析与设计

*说明：本章主要描述，为了实现软件的功能与性能需求，所要采取的整体性的（非局部性的）、高层次的和极其重要的设计策略。与总体结构相关的设计策略，应在第二章中描述，不包含在本章的范围之内。*

## 设计目标与原则

*说明：本小节明确总体功能/性能的设计目标，列出设计时所必须遵循的大原则。*

## 设计策略一：xxxxxx

*说明：本小节与以下各个小节分别说明与总体功能/性能相关的设计策略，每小节各说明一个。设计策略可能涉及的内容包括（但不局限于）：*

## 设计策略二：xxxxxx

# 软件模块说明

*说明：画出软件的模块关系图，并依次对每个模块及模块之间的关系进行说明。*

## 模块关系图

## 模块一：xxxxxx

说明：描述该模块所要实现的主要功能（应该指出完成T3中的哪些USE CASE）以及与其它模块之间的关系。如有必要，可对模块的内部设计和接口设计确定原则性的设计策略。这里描述的设计策略，只需指出总的方向和原则，进一步的细化和设计应留在详细设计阶段完成。

## 模块二：xxxxxx

# 附录1 模块分解一览表

|  |  |  |  |
| --- | --- | --- | --- |
| 模块名称 | 模块编号 | 完成的USE CASE编号 | 模块功能描述 |
|  |  |  |  |

[*该模块编号用于详细设计文档的编号，用于跟踪*]

# 附录2 参考文献

*说明：在这部分列出书写文档所涉及到的文档资料，这些文档通常包括：相关的文档（公司的或者公司以外的）；必备的文档（如体系结构文档）；介绍本文档背景的文档及依据此文档产生的文档（如测试计划文档）等。*

# 修改记录

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 拟制者 | 描述 | 日期 |
| 1.0 | 杨大军 | 修改文档模板编号 | 2000-10-26 |
| 2.0 | 马峥 | 为了使模板更加符合实际操作的需要，对整体的结构和内容进行了调整 | 2002-01-17 |
| 2.0.1 | 杨大军 | 修改了封面和页眉的内容，以便于维护 | 2003-9-9 |
| 2.1 | 杨大军 | 增加模块分解一览表用于上下跟踪 | 2003-11-16 |