

第4章 总体设计方案

4.1 系统概述

4.1.1 项目名称

全名：宜昌市、县、乡、村文档一体化系统

简称：ERMS

版本号：1.1

4.1.2 背景介绍

随着信息化的逐步深入，电子文件的数量与日俱增，电子文件的管理需求也日益迫切，目前的电子文件管理主要存在以下问题。

1. 分散和局部的电子文件管理：目前电子文件管理的功能大多分散于各个业务系统，如办公自动化系统、网站发布系统、邮件系统等，缺乏统一的、全局的覆盖全类型的电子文件管理系统；
2. 不完整的电子文件管理：目前，很多机构的电子文件管理依附于档案管理系统，只涉及了归档文件的电子文件，实质上是电子版归档文件的管理，大量按照现行档案管理办法不需要未归档的电子文件大多散落在业务系统中，甚至在业务承办人员的单机中。另一方面，纳入管理的电子文件的信息也是不完整的，大多只管理文件最终版本，文件关键的过程稿及形成过程缺失；
3. 功能缺失：目前的电子文件管理系统仅包括电子文件的著录、检索、浏览等基本功能；无法和前端业务系统，实现电子文件的自动捕获；无法实现电子文件的综合利用，和后端的电子文件长期保存缺乏衔接，不能做到电子文件生命周期全覆盖。
4. 文件安全性无法保证：目前大部分单位的电子文件管理主要从数据共享角度考虑，对电子文件的真实性、完整性缺乏支持。

4.1.3 主要功能

电子文件管理系统（ERMS）是应用于电子文件形成单位的，旨在捕获电子文件并实施维护、利用和处置的专业系统。它和前段的业务系统、后端的电子文件长期保存系统一起，

实现电子文件的全生命周期管理。

方正电子文件管理系统是依据《电子文件管理系统通用功能需求》(征求意见稿)开发,以内容管理技术为基础,对各级政府文电、应急、文秘、人事等业务系统生成的文件、图片、音像资料等信息资源进行有效的组织、管理、处置、利用的管理系统。通过内容管理技术,方正电子文件管理系统将文本、图像、图形、视音频、网页、邮件等不同类型的电子文件妥善保存,并利用标引、著录、分类、关联等手段对电子文件进行管理、为电子文件的利用和查询提供依据;利用数据分析和挖掘引擎,对数据进行智能化的分析和处理,提供高效的查询手段对所保存的电子文件进行查询和检索,最终使得这些电子文件能够得到最充分的利用,价值不断地提升。

方正电子文件管理系统主要是用来管理机关、团体、企事业单位和其他组织的电子文件,也可以辅助管理非电子的实体文件。主要包括以下内容:

1. OA 系统数据
2. 各种专报、记录等文档
3. 电子期刊、内部网页
4. 公文交换数据
5. 图片、视音频多媒体数据
6. 纸质公文档案

4.1.3.1 电子文件捕获

- a) 支持网络文件夹、手工著录、电子文件接收、自定义接口等多种方式捕获电子文件;
 - i. 网络文件夹:用户可以直接通过业务系统的文件保存功能(如邮件客户端)将文件存储在电子文件管理系统指定的上载区域(Webdav 网络文件夹映射)中,无需先保存在本地再登录系统上传;
 - ii. 手工著录:用户登录系统后上传电子文件后进行文件著录;
 - iii. 电子文件接收:接收从其他电子文件系统移交出来的电子文件;
 - iv. 自定义接口:系统自定义接口,业务系统实现该接口,电子文件管理系统可以定期主动抓取业务系统发布的数据
- b) 自动提取存储或上载文件的元数据,避免重复录入;
- c) 可定制的工作流程,实现捕获电子文件起草的接收、电子文件整理、审核、登记等业务流程;

4.1.3.2电子文件整理

管理员通过分类或检索找到需要处理的电子文件，对电子文件进行元数据补录、分类、组合、关联、授权等操作

- a) 电子文件的元数据及稿本补录、分类、分部门、分文件夹组织；
- b) 电子文件的组合和关联；
- c) 将文本类文件转化版式文件，实现电子文件的固化；
- d) 支持电子公文及各稿本的授权，只有授权的用户才能查看相应文件及相应稿本；

4.1.3.3电子文件统计

- a) 分类别、分部门、分日期进行文件统计；
- b) 支持按自定义条件统计；

4.1.3.4电子文件鉴定处置

- a) 按定义的流程进行电子文件鉴定
- b) 按鉴定结果（续存、销毁、移交）进行处置

4.1.3.5电子文件检索与利用

本功能面向匿名用户和普通用户，根据权限浏览和检索公文，匿名用户可以查看公开的电子文件，登录普通用户可以查看授权的电子文件。主要功能如下：

- 1) 分类浏览电子文件
 - a) 支持不同编目方式浏览电子文件
 - b) 支持按权限浏览电子文件的不同组件、关联、属性等
 - c) 根据按权限下载或列印导出电子文件
- 2) 支持检索结果列印输出或导出电子文件；
 - a) 支持电子文件的元数据检索和全文检索，
 - b) 支持自定义条件和二次检索；
- 3) 利用或订阅申请，对于没有阅读权限的电子文件，可以提出利用申请，按照利用申请流程进行审批，审批完成后后可阅读电子文件。

4.1.3.6文件管理配置

文件管理配置是面向机构的文件管理员，配置电子文件管理系统运行需要的业务数据，目前包括：

- 1) 电子文件类型配置：电子文件类型是具有统一元数据结构的电子文件的集合，系统支持电子文件类型的增加、修改、删除
 - a) 元数据方案：为本类型电子文件配置元数据方案；
 - b) 电子文件视图、著录、检索界面配置：配置并分配各类电子文件的视图、著录、检索界面；用户只能查看、检索被分配视图内的元数据，以保护敏感元数据；
- 2) 电子文件分类方案：配置电子文件的分类方案，电子文件将根据内容放在相应的类目中；
- 3) 电子文件保管期限：配置电子文件的保管期限，每个电子文件都必须具有确定的保管期限，到期后通过鉴定结果确定下一步的续存、销毁和移交操作；
- 4) 存储配置：配置系统的存储服务器及其区域，为各类电子文件 and 用户分配存储服务器和区域；
- 5) 安全配置：配置各类文件缺省的权限配置，授权规则等；
- 6) workflow配置

4.1.3.7安全和系统管理

本功能面向系统管理员，主要配置系统运行需要的用户和参数

- a) 系统配置
- b) 用户管理和功能授权
- c) 日志管理

4.2设计约束

4.2.1需求约束

4.2.1.1标准和规范

- ◇ 本系统遵循国家关于档案管理的有关规定
- ◇ 系统基于 J2EE 架构

✧ 系统利用 HTTP 协议实现 WEB 服务器和客户端通信,通过 FTP 上下载文件、通过 SMTP 发送邮件通知

✧ 本系统是 B/S 系统, 用户界面为 WEB 页面

4.2.1.2 软硬件约束

■ 软件限制

- 功能全部采用 B/S 模式
- 服务器: Windows2008 + Tomcat6.0 + Oracle10g
- 客户端: Windows7 + IE7

■ 硬件限制

- 电子文件管理服务器: 双 CPU, 4G 内存

4.2.1.3 用户界面约束

使用蓝色界面风格, 同时有一个方正公司 LOGO

4.2.1.4 软件质量约束

在满足目前管理的实际需求基础上, 适度考虑系统的前瞻性。系统应在统一规划的原则下, 满足开放性和高可扩展性、易用与易维护性、安全可靠性的要求。

性能要求:

- 并发用户数, 系统支持 100 个以上并发用户数;
- 查询性能: 常规服务器配置下, 在 100 万以下资产库内, 单用户查询返回首页费时 < 3 秒, 并发用户数 > 50 时返回首页查询时间 < 5 秒;

对更高数据量的查询性能, 暂时不提出性能要求, 但必须测试出性能数据;

■ 可靠性要求。

- 关键程序和数据均有足够的备份、容灾措施。
- 系统有效工作时间 $\geq 99\%$, 系统故障平均间隔时间 ≥ 90 天。

■ 扩展性要求

- 功能扩展

要求考虑多单位集中使用电子文件管理系统问题, 要求系统具有良好的功能扩展性、平台性和自定义性。

- 容量扩展

随着信息资源的增加，系统容量的扩展不应涉及系统的再次开发，而只是改变有关设置参数。容量扩充时，原有信息资源应不受影响且能平滑迁移。

■ 通用性要求

能根据用户要求定制数据结构，包括打印报表、编目格式、档案分类。

■ 易用性要求

突出智能化、自动化的特点，支持对大量电子文件数据的自动化处理，减少管理人员的工作量，提高工作效率。为电子文件用户提供智能化的查询利用工具，提高电子文件的可用性。

■ 先进性要求

采用 B/S 模式，支持网络环境下的多用户并发操作。

4.2.2 隐含约束

本软件基于 Apache Struts MVC Framework 构建，通过 Struts2.0 Framework，将软件分为 Model（模型）、View（视图）、Controller（控制）3 层，View 层负责与用户的界面及其交互，Controller 层负责程序控制，将用户的操作转交给模型层，并转换到下一个 View，Model 层负责应用及数据模型。

本软件基于博思内容管理平台 4.0 构建，系统通过博思的开发接口，调用博思的相应功能，简化系统的开发工作。

4.3 设计策略

本系统在设计策略上有以下考虑：

- ◇ 扩展策略：采用接口和向下兼容的策略，隔离调用和具体的实现。
- ◇ 复用策略：采用组件的策略，以重用所开发的组件。
- ◇ 高效策略：服务器返回客户端的交换格式采用 JSON 格式，减少网络的开销，提高系统效率。
- ◇ 先进策略：融入全文检索、基于 B/S 架构、基于 J2EE 体系规范、基于 XML 数据交换标准。
- ◇ 现实策略：实现手工管理与计算机化管理相结合。

4.4 程序系统体系结构

4.4.1 总体功能设计

宜昌市、县、乡、村文档一体化系统，整个系统采用 B/S 架构，既支持集中式部署，也支持分布式部署模式。属于电子文件全生命周期的中间系统，处于各业务应用系统与长期保存管理系统之间，具有承上启下的作用。

为了实现电子文件的收集与归档，电子文件管理信息系统在制订统一电子文件标准和操作流程规范的基础上，需要实现相关部门、各机构电子文件的分类、格式、标识、登记与著录，通过标准的数据接口规范和业务流程规范实现各类业务系统和电子文件管理信息系统的数据交换与共享，形成各类电子文件，并对电子文件的内涵进行充分揭示，以支持电子文件的高效利用。因此系统需提供各种接口来捕获各种业务系统的电子文件，实现电子文件的集中管理和利用，消除信息孤岛。

系统需要提供对捕获过来的电子文件进行补录、分类整理、案卷整理功能，保证电子文件的利用和管理，也为了与长期保管系统的无缝衔接。系统需要记录电子文件的所有操作过程及操作过程中形成的电子文件，实现对电子文件的全生命周期的管理。

从以上分析来看，电子文件管理系统的业务流程如下图所示。

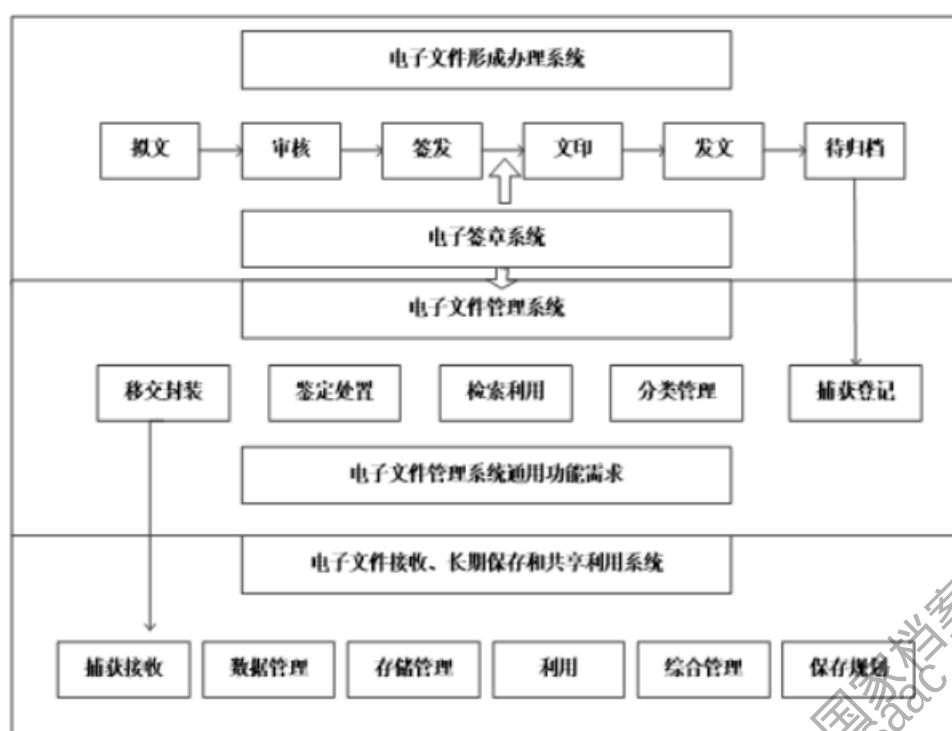


图 3.1 电子文件管理系统业务流程

系统业务流程是从 OA 系统或别的系统捕获电子文件，经过整理、分类、补录等整理过程形成电子文件，提交审核，审核通过之后电子文件正式存档，成为有价值可利用的电子文件，利用人员通过调阅、订阅、收藏之后获得电子文件，电子文件有自己的生命周期，到一段时间之后需对其进行续存、销毁、移交等处置，处置需要鉴定小组批准之后才能操作。需要归档的电子文件, 参照电子文件归档管理办法, 按照相关标准, 进行移交, 归档, 保存。.

4.4.1.1 总体功能设计

通过以上对电子文件管理系统的需求分析，电子文件管理系统功能主要包含有文件管理配置方面、文件管理业务功能方面、安全管理方面，以及系统管理方面的功能。功能架构如图 3.2 所示：

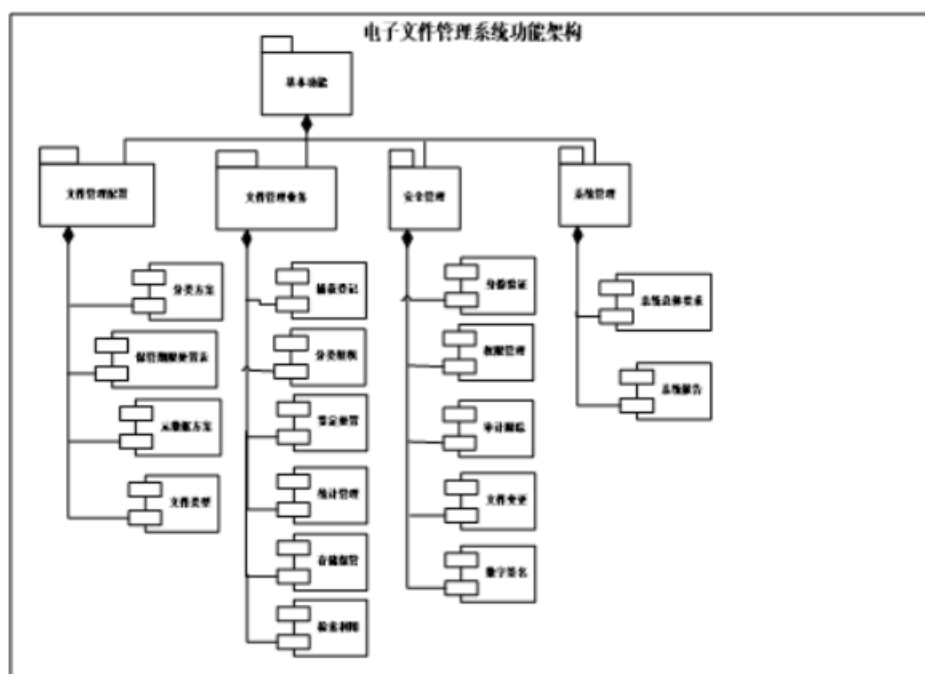


图 3.2 电子文件管理系统功能模块示意图

其中，文件管理配置功能是电子文件管理系统中建立和维护文件管理业务规范的功能，包括文件类型管理、元数据管理等；文件管理业务功能主要基于电子文件管理业务流程展开，包括捕获、分类、检索等；安全管理功能是保护电子文件以及电子文件管理系统安全的功能；系统管理功能是指电子文件管理系统运行所需要的基本环境支撑、工具支撑等内容。

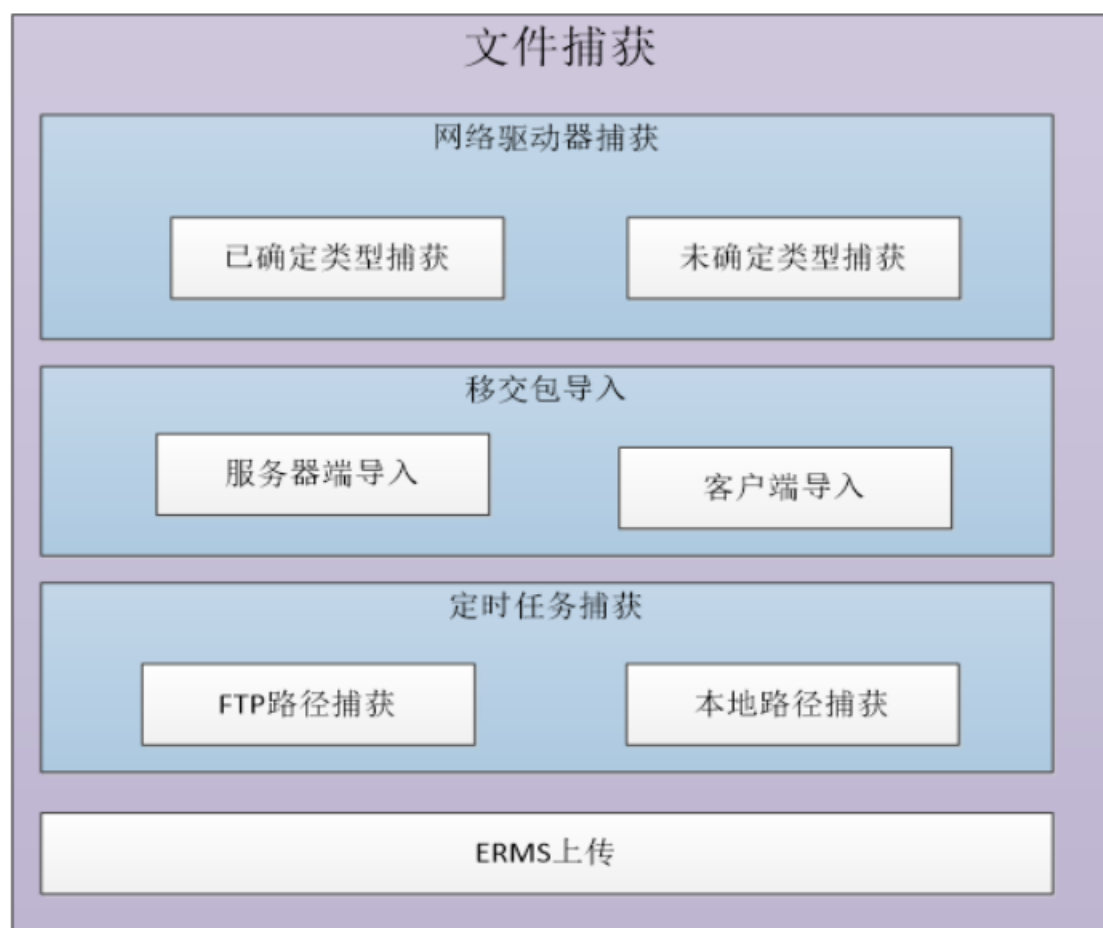
4.4.1.2电子文件捕获

用于对外部系统发送的文件进行接收，是保障电子文件管理系统数据来源的唯一入口，能够通过多种途径捕获各种类型、多种格式的电子文件数据。本次开发的电子文件捕获渠道总共有两种：

1) 手工登记：文件管理员手工录入电子文件。

2) 自动捕获：提供标准的集成接口和应用系统集成，自动抓取应用系统产生的电子文件。

本系统实现采用数据接口，采用 zip 结构，zip 文件由一个描述信息的 xml 文件和一个或多个组件组成。



4.4.1.3电子文件分类组织整理

电子文件整理主要包括分类整理和案卷整理功能，能够对整理后的电子文件进行元数据著录、数据关联、分类调整、案卷调整、业务行为等信息的处理及显示等操作。

文件整理：对捕获过来的电子文件进行补录，建立关联等。

分类整理：对入库的电子文件依据分类方案进行增删改的操作，并且能够进行拆分、

移动、复制、合并等操作。系统的分类管理人员能够对电子文件进行元数据补录,按照分类、部门、文件夹的结构进行组织,文件之间可进行组合、关联、授权等操作。

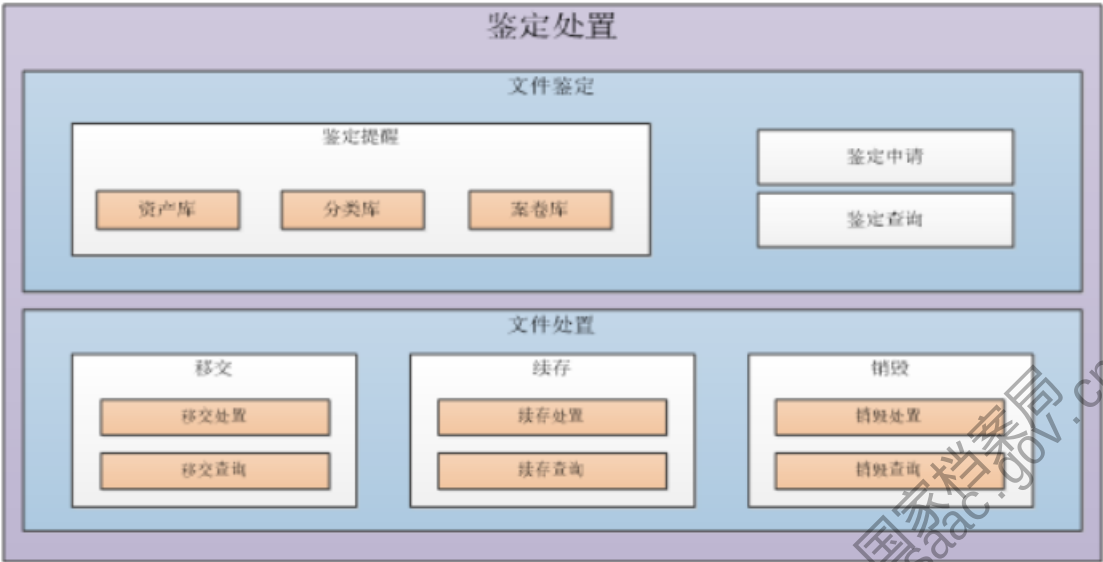


图 3.3 电子文件整理功能模块图

案卷整理：依据案卷分类表和案卷关联表对捕获的电子文件进行增删改的操作，并且能够对案卷进行合并、复制和拆分等。

4.4.1.4 电子文件鉴定处置

整理过后的电子文件需要进行鉴定处置，鉴定过程是通过系统自动实现的。鉴定内容主要分为两类，即文件到期鉴定和文件开放鉴定。



文件到期鉴定主要是针对到期的文件提交鉴定申请，以便对到期的文件做处置，处置结果分为移交、销毁和续存三种。文件开放鉴定是针对文件需要开放，以便大家都可以查询的鉴定处置

图 3.4 电子文件鉴定功能示意图

文件到期提醒：系统依据《保管期限与处置表》制定的鉴定策略，自动提醒文件管理员到期需要鉴定的文件，并以列表显示。

文件鉴定申请：文件管理员依据鉴定管理流程发起鉴定工作，将到期的电子文件以列表的形式提出申请。

文件鉴定审核：鉴定审核人员进行审核处置。

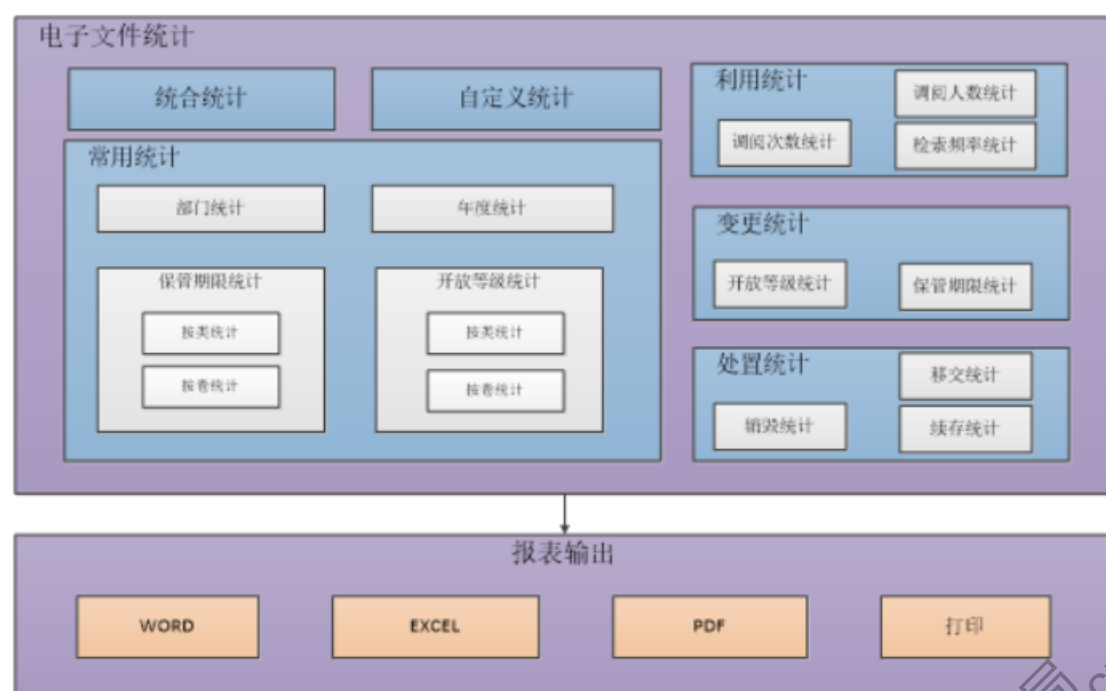
文件鉴定结果查询：与流程相关的工作人员可随时查询审核的过程和结果。

文件移交处置：根据鉴定结果做移交的处置。

文件销毁处置：如鉴定结果为销毁，系统将待销毁的电子文件进入销毁流程，即销毁申请、审核、确认，并可查询其处理状态。

文件续存处置：根据鉴定结果做续存的处置。

4.4.1.5 电子文件统计分析



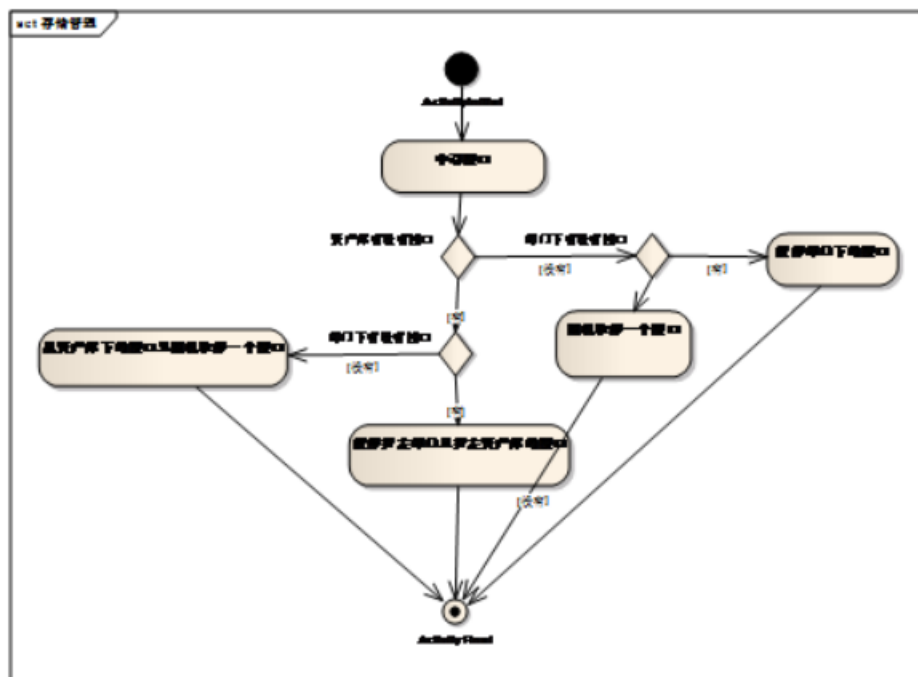
电子文件统计分析的功能是针对应用系统中电子文件的存量状态和业务处置的过程或结果进行不同条件下的统计分析。主要应用在统计及统计后形成个性化需要的报表，并通过各种图形化的形式展现。

4.4.1.6电子文件存储管理

电子文件管理系统最重要的应用目标是对各种类型的电子文件进行海量的数据存储,系统需要有一定的安全性、可靠性,以及可扩展性。

根据电子文件存储需求,在系统中设计了以下几种存储策略:

根据电子文件存储的需求，系统采用“集中管理，分布存储”的方式对电子文件进行存储管理，集中管理是指元数据的集中管理，分布存储是指电子文件的分布存储，电子文件可以存储在不同的服务器上，比如不同区域有不同的存储服务器，用户访问电子文件时可以就近访问电子文件，提高访问效率，减少服务器的网络压力。存储系统支持 HTTP、FTP 等标



准协议,可以按照电子文件类型分配,也按用户机构分配存储。存储接口的访问规则如下图所示。

图 3.5 电子文件存储接口访问流程图

文件上传的时候需要根据上传用户所在的部门和上传的文件库来获得文件的上传存储接口。

● 关系型数据存储

在系统中电子文件的元数据存储是通过关系数据库存储的，目前支持主流的多种数据库。

- 索引数据存储

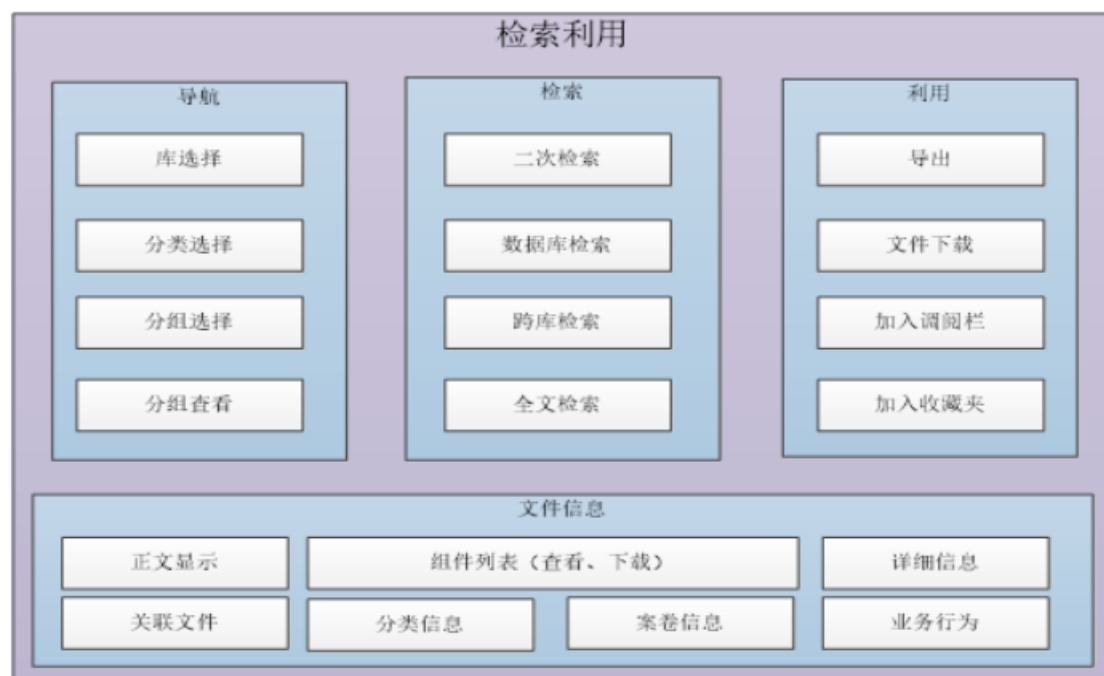
为便于全文检索的利用，系统设计了索引数据的存储，便于提高检索效率，方便增量检索数据的存储。该存储策略，描述了存储的数据是逻辑集中，物理分布的。通过分级缓存、就近访问的策略提高存储和访问的效率。

4.4.1.7 电子文件检索利用

检索功能模块：包括跨文档库查询、全文检索、高级检索、模糊检索、在查询结果集中二次检索、检索条件项的用户定制、分类检索等功能。

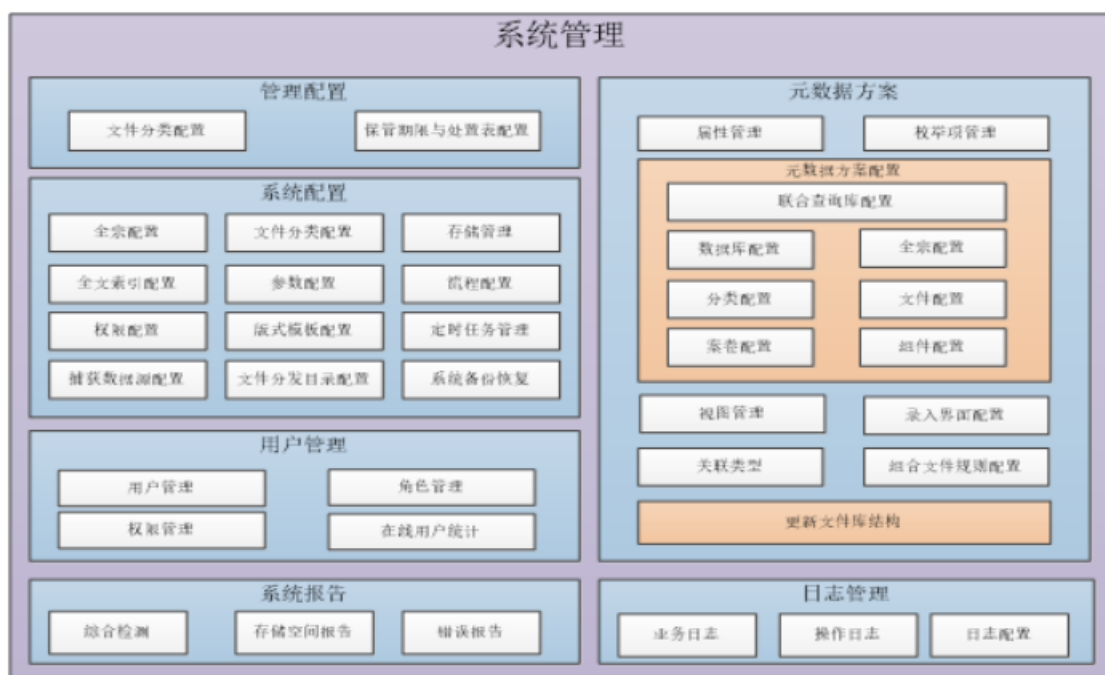
为保证系统的安全，在检索查询的过程中，需要根据检索的人员的身份及权限判断检索查询范围。

系统根据权限判断检索的结果，能否在同一个界面上支持用户查看电子文件原件、目录、OCR 识别后的文字部分、关联文件信息、文件的附件信息等，并将检索结果集成在同一个界面上实现文档查询的便利。



4.4.1.8 文件配置管理

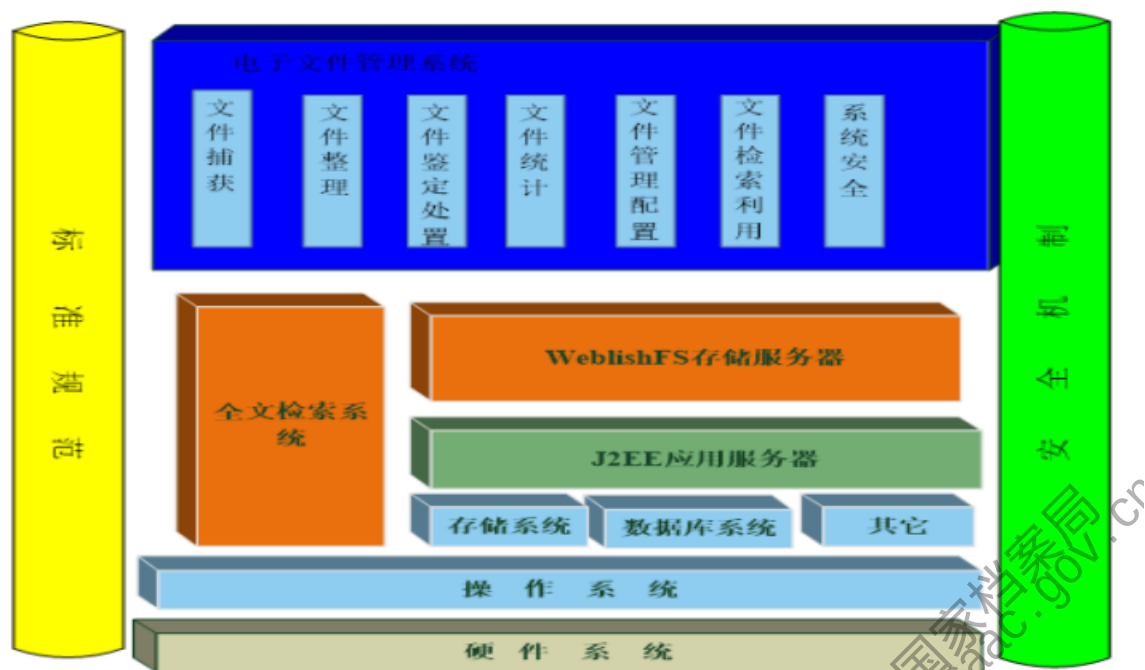
电子文件分类方案是电子文件管理业务的基础，支持用户建立和维护符合自身实际的分类方案，依据分类方案对文件进行的标识，并按照文件的特定类型进行形成、办理、流转及归档管理。



4.4.2 总体层次结构

方正电子文件管理信息系统的总体结构由多层组成，最底层为硬件系统和操作系统，其上依次为数据库服务器、J2EE 应用服务器、在此基础上，建立起电子文件的管理平台。管理平台往下有全文检索和文件服务器来支撑。

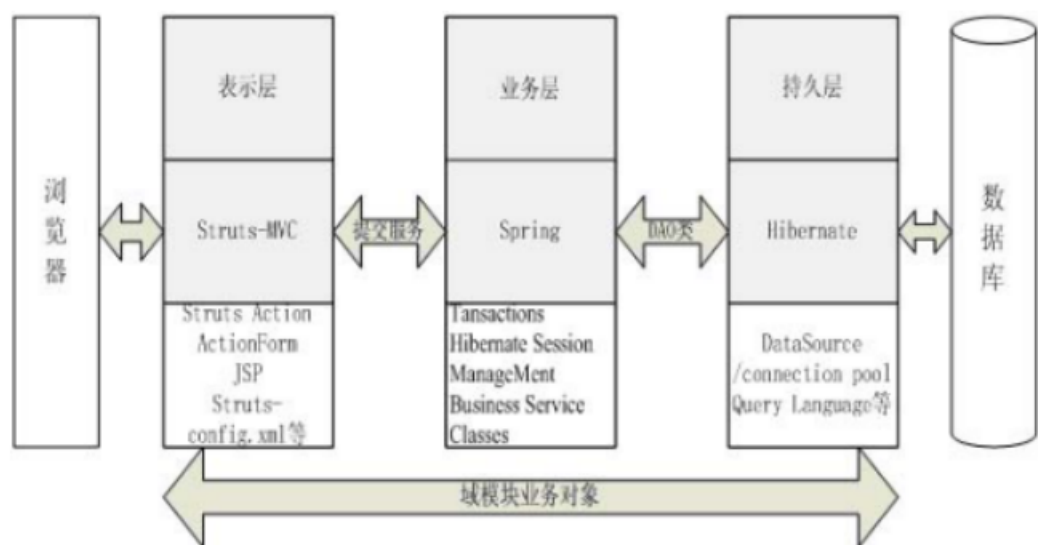
系统总体结构的示意图如下：



图：系统总体结构示意图

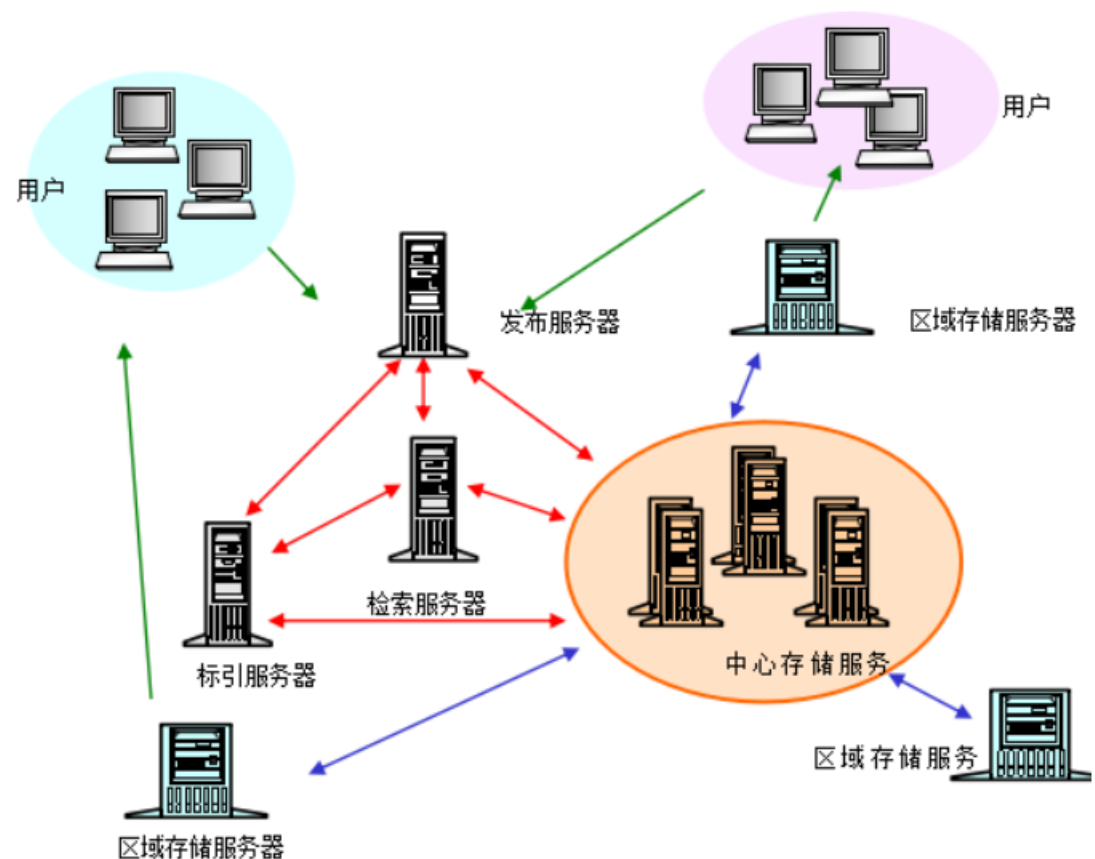
在系统设计中，始终以标准规范和安全机制作为支撑系统的两大支柱。标准性体现在技术标准性和业务标准性这两个方面，在技术上，通过采用 XML、Java/J2EE、OAI 等国际标准和 ERMS 标准，保证系统的先进性；在业务上，通过采用可定制性基础数据和业务流程，满足系统具有足够的灵活性和扩展性。

4.4.3 系统体系架构



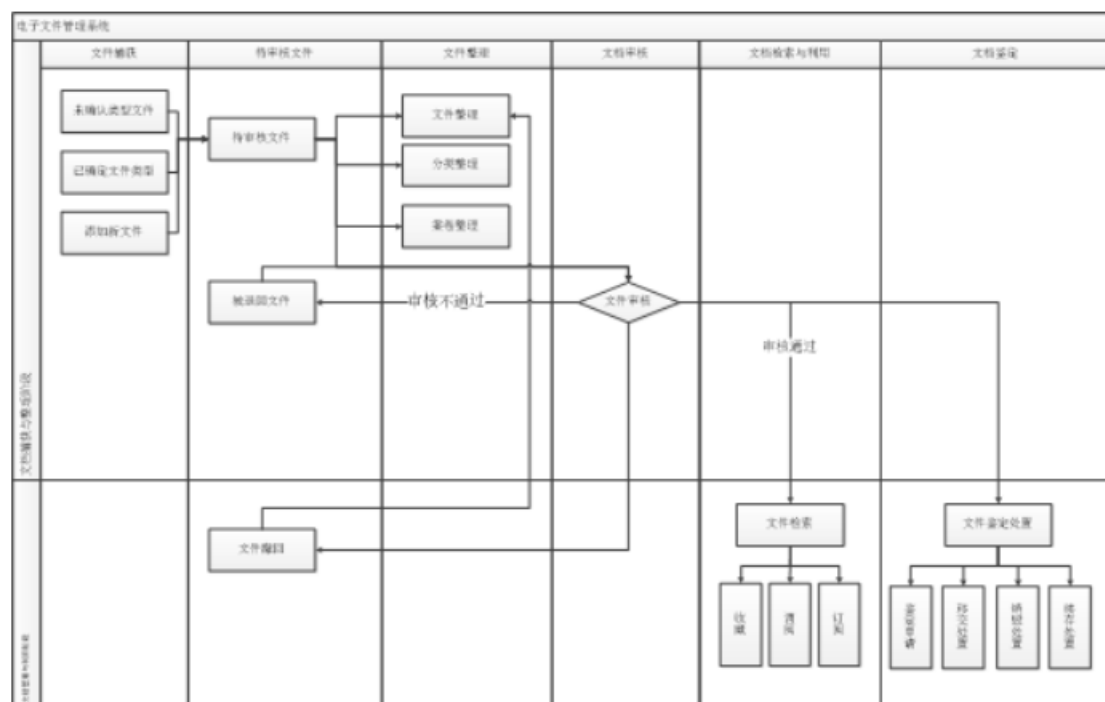
框架层次结构概述框架层次结构层次分为四层：表现层、持久层、业务层、域对象层。每个框架分别负责一个层次的实现，表现层我们使用 Struts，业务层使用 Spring，持久层使用 Hibrenate。每层在应用程序中都有明确的责任。每一应用层彼此独立且保持一致。这些框架层次在以一种松散耦合的方式彼此作用而不用管低层的技术细节。

4.4.4部署模型图



4.4.5总体业务流程

整理业务流程是从 OA 系统或别的系统捕获电子文件，经过整理、分类、补录等整理过程形成电子文件，提交审核，审核通过之后电子文件正式存档，成为有价值可利用的电子文件，利用人员通过调阅、订阅、收藏之后获得电子文件，电子文件有自己的生命周期，到一段时间之后需对其进行续存、销毁、移交等处置，处置需要鉴定小组批准之后才能操作。



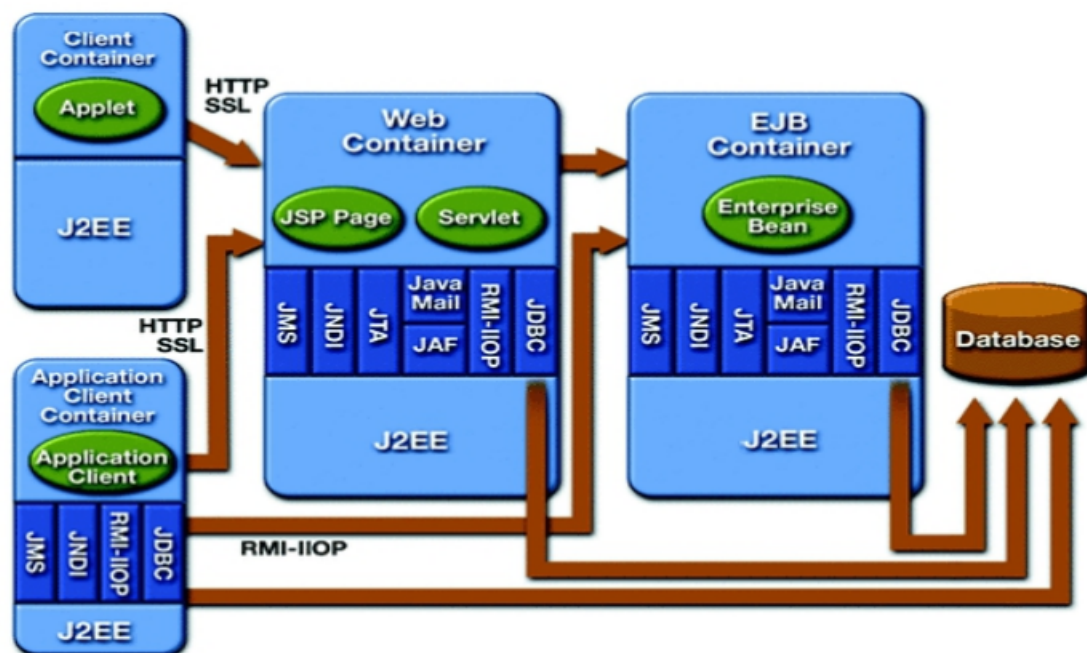
4.4.6 关键技术与算法

4.4.6.1 Java/J2EE 跨平台技术

运用目前业界先进的基于 Java/J2EE 技术，信息平台成为真正意义上的跨平台的系统。基于标准 J2EE 规范开发，运用丰富的设计模式（工厂模式、单态模式等），采用 Java Bean 的业务逻辑封装，采用 JSP/Servlet 的表现逻辑设计。

J2EE 平台提供了一个基于组件的方法，来设计、开发、装配及部署企业应用程序。J2EE 平台提供了多层的分布式的应用模型、组件再用、一致化的安全模型以及灵活的事务控制。软件提供商不仅可以比以前更快的速度向市场推出创造性的客户解决方案，而且，这种平台独立的、基于组件的 J2EE 解决方案不会被束缚在任何一个厂商的产品和 API 上。

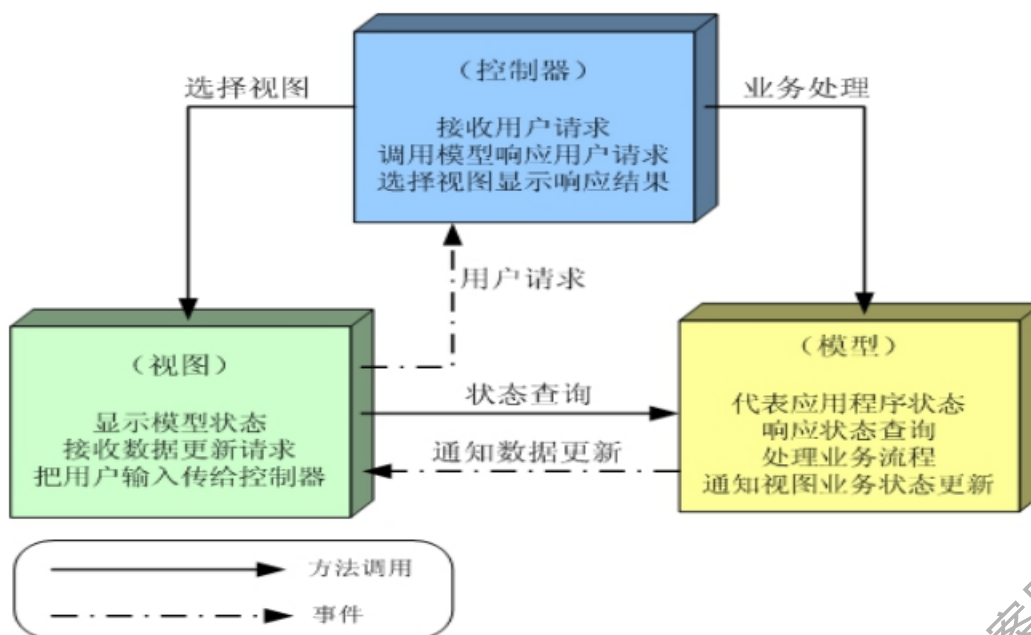
下图是 J2EE 标准组件及其相互关系的示意图：



图：J2EE标准组件及其相互关系示意图

4.4.6.2 MVC 模式设计

本系统各单元在设计上均符合 MVC 模式，把程序分为三个核心模块：模型、视图和控制器，分别负担不同的任务。下图显示了这几个模块各自的功能以及它们的相互关系：



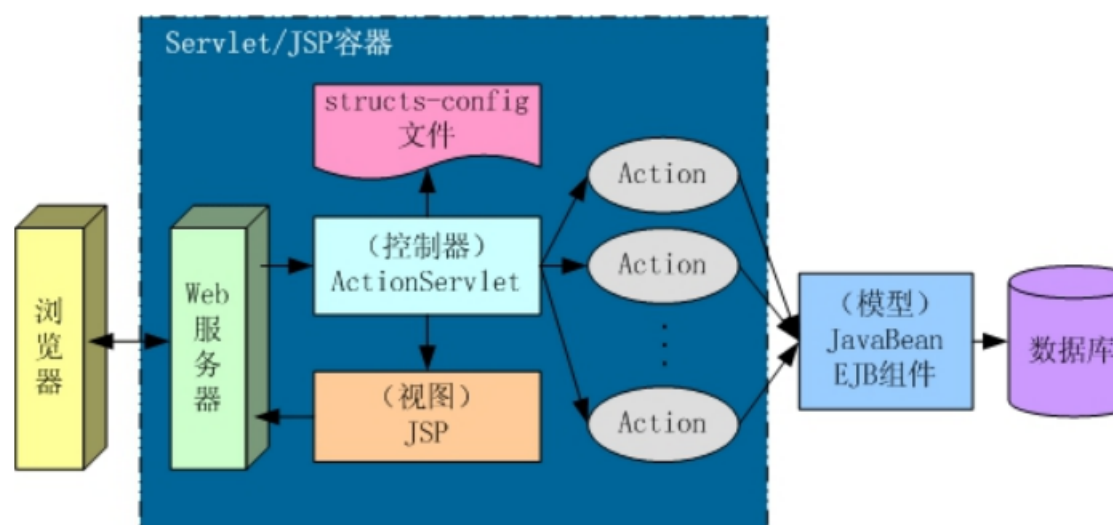
图：MVC设计模式

4.4.6.3 Struts 框架技术

信息平台基于 Apache Struts MVC 框架构建。Struts 框架技术可以大大简化 Web 应用的开发过程，提高开发效率，缩短开发周期，使得开发人员可以把精力集中在如何解决实际业务问题上。

本系统使用 Struts 2.0 框架来实现 MVC 模式。通过 Struts 框架，将软件分为 Model（模型）、View（视图）、Controller（控制）3 层。

View 层由一组 JSP 文件构成，负责与用户的界面及其交互；Controller 层由 Action Servlet 和 Action 来实现，负责程序控制，将用户的操作转交给模型层，并转换到下一个 View；Model 层由实现业务逻辑的 JavaBean 或 EJB 组建成，负责应用及数据模型。下图显示了 Struts 实现的 MVC 框架：



图：Struts实现MVC框架

4.4.6.4 XML 技术

XML 是 eXtensible Markup Language（可扩展置标语言）的简称，出现于 1996 年，并于 1998 年成为 W3C 正式推荐标准。XML 具有一系列优点，主要表现在以下几个方面：

- 良好的可扩展性
- 内容与形式的分离
- 遵循严格的语法要求
- 便于不同系统之间信息的传输

■ 具有较好的保值性

信息平台的构建以 XML 技术为基础:

- 基于 XML 格式数据的导入/导出
- 基于 XML 格式的系统配置
- 基于 XML 格式数据交换, 内部的模块级信息交换
- 基于 XML 格式的二次开发接口定义, 接口级信息交换都使用了 XML 信息格式

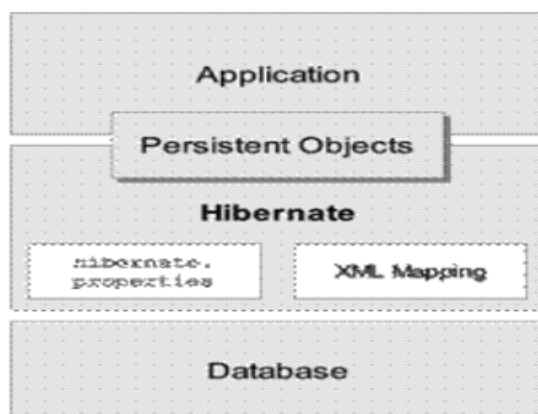
4.4.6.5 内容检索技术

方正电子文件管理系统提供了多种检索手段, 可实现跨库检索。检索内容, 检索方式可订制, 让用户可以快捷、准确地找到想要的资产内容。系统提供元数据检索、全文检索等等。搜索结果可以以收藏夹的方式保存起来, 方便用户以后再次利用。

4.4.6.6 Hibernate 技术

Hibernate 是一个开放源代码的对象关系映射框架, 它对 JDBC 进行了轻量级的对象封装, 使 Java 程序员可以随心所欲的使用对象编程思维来操纵数据库。它不仅提供了从 Java 类到数据表之间的映射, 也提供了数据查询和恢复机制。相对于使用 JDBC 和 SQL 来手工操作数据库, Hibernate 可以大大减少操作数据库的工作量。另外 Hibernate 可以利用代理模式来简化载入类的过程, 这将大大减少利用 Hibernate QL 从数据库提取数据的代码的编写量, 从而节约开发时间和开发成本。Hibernate 可以和多种 Web 服务器或者应用服务器良好集成, 如今已经支持几乎所有的流行的数据库服务器。

Hibernate 技术本质上是一个提供数据库服务的中间件。它的架构如图所示:



上图显示了 Hibernate 的工作原理, 它是利用数据库以及其他一些配置文件如 Hibernate .properties , XML Mapping 等来为应用程序提供数据持久化服务的。

Hibernate 具有很大的灵活性,但同时它的体系结构比较复杂,提供了好几种不同的运行方式。在轻型体系中,应用程序提供 JDBC 连接,并且自行管理事务,这种方式使用了 Hibernate 的一个最小子集;在全面解决体系中,对于应用程序来说,所有底层的 JDBC/JTA API 都被抽象了, Hibernate 会替你照管所有的细节。

方正电子文件管理采用 Hibernate 技术实现对数据层的操作。

4.4.6.7 Spring 技术

Spring 是一个开源框架,是为了解决企业应用程序开发复杂性而创建的。框架的主要优势之一就是其分层架构,分层架构允许您选择使用哪一个组件,同时为 J2EE 应用程序开发提供集成的框架。

Spring 框架是一个分层架构,由 7 个定义良好的模块组成。Spring 模块构建在核心容器之上,核心容器定义了创建、配置和管理 bean 的方式,如图 1 所示。

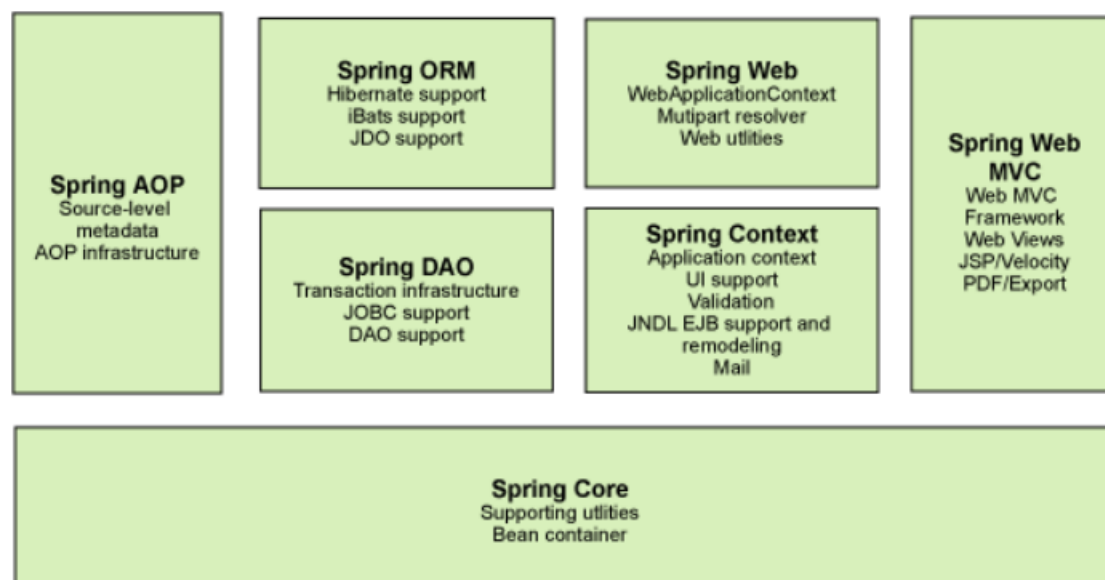


图 1. Spring 框架的 7 个模块

组成 Spring 框架的每个模块(或组件)都可以单独存在,或者与其他一个或多个模块联合实现。每个模块的功能如下:

- **核心容器:** 核心容器提供 Spring 框架的基本功能。核心容器的主要组件是 BeanFactory,它是工厂模式的实现。BeanFactory 使用控制反转 (IOC) 模式将应用程序的配置和依赖性规范与实际的应用程序代码分开。

- **Spring 上下文:** Spring 上下文是一个配置文件，向 Spring 框架提供上下文信息。Spring 上下文包括企业服务，例如 JNDI、EJB、电子邮件、国际化、校验和调度功能。
- **Spring AOP:** 通过配置管理特性，Spring AOP 模块直接将面向方面的编程功能集成到了 Spring 框架中。所以，可以很容易地使 Spring 框架管理的任何对象支持 AOP。Spring AOP 模块为基于 Spring 的应用程序中的对象提供了事务管理服务。通过使用 Spring AOP，不用依赖 EJB 组件，就可以将声明性事务管理集成到应用程序中。
- **Spring DAO:** JDBC DAO 抽象层提供了有意义的异常层次结构，可用该结构来管理异常处理和不同数据库供应商抛出的错误消息。异常层次结构简化了错误处理，并且极大地降低了需要编写的异常代码数量（例如打开和关闭连接）。Spring DAO 的面向 JDBC 的异常遵从通用的 DAO 异常层次结构。
- **Spring ORM:** Spring 框架插入了若干个 ORM 框架，从而提供了 ORM 的对象关系工具，其中包括 JDO、Hibernate 和 iBatis SQL Map。所有这些都遵从 Spring 的通用事务和 DAO 异常层次结构。
- **Spring Web 模块:** Web 上下文模块建立在应用程序上下文模块之上，为基于 Web 的应用程序提供了上下文。所以，Spring 框架支持与 Jakarta Struts 的集成。Web 模块还简化了处理多部分请求以及将请求参数绑定到域对象的工作。
- **Spring MVC 框架:** MVC 框架是一个全功能的构建 Web 应用程序的 MVC 实现。通过策略接口，MVC 框架变成高度可配置的，MVC 容纳了大量视图技术，其中包括 JSP、Velocity、Tiles、iText 和 POI。

Spring 框架的功能可以用在任何 J2EE 服务器中，大多数功能也适用于不受管理的环境。Spring 的核心要点是：支持不绑定到特定 J2EE 服务的可重用业务和数据访问对象。毫无疑问，这样的对象可以在不同 J2EE 环境（Web 或 EJB）、独立应用程序、测试环境之间重用。

方正电子文件管理系统采用 Spring 的 ICO 管理系统对象的生成，采用 Spring 的 AOP 技术实现数据库事务的管理。

4.4.7 关键数据结构

4.4.7.1 全局变量说明

将系统使用到的常量定义到指定的文件中，类名定义为：

com.founder.irms.commons.const.Constant.java, 该文件中的值是静态的, 可以在程序其他地方直接引用。包括以下内容:

- 资产状态的定义
- 资产库 ID 的定义
- 左侧模块菜单的定义
- 其他常量定义

一、资产状态值

组件的状态

常量名	常量值	用途	对应数据库字段
STATUS_COMPONENT_NO_RECIEVE	6	待接收	app_N_file中status字段
STATUS_ COMPONENT _ RECIEVED	5	已接收	app_N_file中status字段
STATUS_ RECORDS _NO_RECIEVE	6	待接收	app_N_asset中status字段
STATUS_ RECORDS _ RECIEVED	5	已接收	app_N_asset中status字段
STATUS_ RECORDS _NO_INSPECT	4	待审核	app_N_asset中status字段
STATUS_ RECORDS _ RETURNED	3	已退回	app_N_asset中status字段
STATUS_ RECORDS _ WITHDRAWED	2	已撤回	app_N_asset中status字段
STATUS_ RECORDS _ WITHDRAWED	0	已审核	app_N_asset中status字段
STATUS_ RECORDS _NO_SUMMIT	2	待提交	app_50_asset(调单库)中status字段
STATUS_LEND _SUMMITED	1	已提交	app_50_asset(调单库)

			中status字段
STATUS_LEND _SUMMITED	0	已审核	app_50_asset(调单库) 中status字段
STATUS_ APPRAISAL _NO_SUMMIT	2	待提交	app_52_asset(鉴定库) 中status字段
STATUS_ APPRAISAL _SUMMITED	1	已提交	app_52_asset(鉴定库) 中status字段
STATUS_ APPRAISAL _INSPECTED	0	已审核	app_52_asset(鉴定库) 中status字段
STATUS_CLASSFICATION_CLOSE	1	已关闭	app_3_asset(分类表)中 status字段
STATUS_CLASSFICATION_OPEN	0	已开放	app_3_asset(分类表)中 status字段

二、 资产库 ID 的定义

需要初始化的资产库及常量说明如下：

1) 配置型资产库：

常量名	常量值	用途
REP_CLASSFICATION	2	分类方案表
REP_CLASS	3	分类表
REP_GAOBEN	4	稿本库
REP_ FONDS	5	全宗表

2) 应用型的资产库

常量名	常量值	用途
REP_LEND	50	调阅表
REP_SUB	51	订阅表
REP_INDETIFY	52	鉴定表
REP_ DESTORY	53	销毁表
REP_ FAVORITE	54	收藏夹表

REP_BASKET	55	借阅篮表
------------	----	------

3) 资源型的资产库

常量名	常量值	用途
REP_DOCUMENT	100	文档库
REP_FILE	101	案卷库
REP_IMAGE	102	图片库
REP_EMAIL	103	邮件库

```

public class Constant {
    public Constant() { }

    //模块菜单定义
    public static final intSTATUS_COMPONENT_NO_RECIEVE=6;
    public static final intSTATUS_ COMPONENT _ RECEIVED=5;
    public static final intSTATUS_ RECORDS _NO_RECIEVE=6;
    public static final intSTATUS_ RECORDS _ RECEIVED=5;
    public static final intSTATUS_ RECORDS _NO_INSPECT=4;
    public static final intSTATUS_ RECORDS _ RETURNED=3;
    public static final intSTATUS_ RECORDS _ WITHDRAWED=2;
    public static final int STATUS_ RECORDS _ WITHDRAWED=0;
    //其他常量的定义，可以随时补充

```

4) 左侧模块菜单的定义

左侧导航菜单的显示采用XML配置，系统启动的时候加载到缓存里，也可以系统管理的加载数据结构里重起加载菜单项，其设计参考第六章模块菜单设计。

4.4.7.2数据交换格式

方正文件管理系统上传文件时需要自动提取文件的属性，文件的属性通过文件服务器（WEBLISHFS）获取，系统与文件服务器之间的数据交换格式为XML格式，例子如下：

uploadInfoXml 示例

```

<?xml version="1.0" encoding="utf-8" ?>

<File>

<UploadUser>cmsuser</UploadUser>

<UploadTime>2011 09 22 15:08:25</UploadTime>

<UploadClient>fe80:0:0:0:6007:6083:732a:a8e3(fe80:0:0:0:6007:6083:732a:a8e3)
</UploadClient>

<UploadApplication>Microsoft-WebDAV-MiniRedir/6.1.7601</UploadApplication>

<ContentType>application/octet-stream</ContentType>

<ContentLength>14433</ContentLength>

<Attachments></Attachments>

<AttachmentsEmbedded>true</AttachmentsEmbedded>

<Signature>hYIMRvwGOBzo8VPMRxKehkwc/FswNB3CpNtOKonVu+ScOrtqiow9P99IKmBN50Dtb
6fv5tjmqi6QYig7GTtnS/WqS00/EKOM7fRU/j35vpYzihgBrb50UUUs5+CAF7n29V69+D7c0nasVqqKc
u+CL+z1rdkNqWisFRh713DL+0g=</Signature>

</File>

```

weblish:document 对应的 buildInPropertiesXml 示例

```

<?xml version="1.0" encoding="utf-8" ?>

<Properties xmlns="weblish:document">

<Application>Microsoft Office Word 12.0000</Application>

<Author>Wang Xusheng</Author>

<CreatedTime>2011 09 22 15:08:00</CreatedTime>

<Creator/><Description/><Identifier/>

<Keywords/>

<LastModifiedTime>2011 09 22 15:08:00</LastModifiedTime>

<PageCount>1</PageCount>

<Subject/>

<Title/>

</Properties>

```

weblish:message 对应的 buildInPropertiesXml 示例

```

<?xml version="1.0" encoding="utf-8" ?>
<Properties xmlns="weblish:message">
<BCC/>
<CC/>
<From>LangYanRui(郎彦睿)
<O=FOUNDERHOLD/OU=FOUNDERHOLD/CN=RECIPIENTS/CN=LANG_YANRUI></From>
<Identifier><6B877440AD196046980E23057C6FA79E03A91B17@mail03. HOLD. FOUNDER. COM></
Identifier>
<ReceivedTime/><SentTime>2011 09 22 15:05:26</SentTime>
<Title>转发: 9 月 24 日北大人民医院参观和公司交流安排--0922</Title>
<To>WangXusheng(王绪胜) <wx@FOUNDER. COM. CN></To>
</Properties>

```

可能扩展的命名空间还包括 weblish:image, weblish:webpage。

其中 xmlns 是命名空间, weblish:document 表明是录入文档库, 通过资产库的命名空间属性去对应, 系统有一个全局的配置变量去决定是否根据命名空间来确定资产库类型。其余的属性需要与系统建同样的字段去对应, 且大小写一样。

4.4.7.3 数据库设计

1. 数据库物理设计

为提高数据库访问性能, 充分利用数据库表空间, 表空间物理文件可以部署到不同的物理磁盘或 RAID 盘阵, 以降低读取和写入数据时对存储空间的竞争。

2. 数据库外部设计

- 数据库服务器支持 RAC, Oracle10g 在集群中两个节点之间自动平衡用户负载及预防单点故障。
- 数据库服务器的部署应同应用服务器在同一局域网内, 且最好在同一 domain 之内
- 数据库服务器同应用服务器在逻辑上分离, 为保证性能建议实现物理分离

3. 对象命名约定

1. 系统表以 sys 为前缀，应用表以 app 为前缀，前缀与后续名称前以下划线分隔。如：系统表中的存储服务器配置表 sys_StorageServer，应用表中的资产基本属性表 app_asset_1。一个资产库生成的表名如下：

资产基本信息表	app_asset_{资产库 ID}
资产库和多值属性关系表 (N)	app_asset_{资产库 Id}_{属性 Id}
资产包含文件表	app_asset_file_{资产库 Id }
资产关联关系表	app_associate_{资产库 Id }
资产被关联关系表	app_associated_{资产库 Id }
资产授权表	app_asset_authority_{资产库 Id }
资产 ACL 表	app_asset_acl_{资产库 Id }

2. 索引命名：如果该表只有一个索引，则采用 “<表名>_INDEX”；如果有存在多个索引，则采用 “<表名>_<字段名>_INX”。数据库对象大小写为首字母大写，其他小写，缩略词全部大写。如：字段名 ResourceID。

避免使用数据库保留字。如：view、date、level、number、read、execute、left、right、width、height。

4.5 公共模块及公共类

4.5.1 包名定义

除了博思提供的 cm 包外，系统中新加的类都需要遵循以下包的存放规则。

主要包名如下：

包名	作用
Com.founder.erms.common.web	放置公共模块的 Action 类
Com.founder.erms.common.entity	放置公共模块的实体类
Com.founder.erms.common.service	放置公共模块的业务类
Com.founder.erms.common.dao	放置公共模块的数据访问类 (hibernate)
Com.founder.erms.common.model	放置公共模块的实体类 (hibernate)
Com.founder.erms.common.consts	放置公共模块的常量

Com. founder. erms. commons. util	放置公共模块的公共类
Com. founder. erms. 模块名. web	放置具体模块的 Action 类
Com. founder. erms. 模块名. entity	放置具体模块的实体类
Com. founder. erms. 模块名. service	放置具体模块的业务类
Com. founder. erms. 模块名. dao	放置具体模块的数据访问类 (hibernate)
Com. founder. erms. 模块名. model	放置具体模块的实体类 (hibernate)
Com. founder. erms. 模块名. consts	放置具体模块的常量
Com. founder. erms. 模块名. util	放置具体模块的公共类

主要模块包名说明如下:

admin	管理模块
flow	流程模块
user	用户管理模块
lend	借阅订阅模块
myspace	我的工作平台模块
capture	捕获模块
statistics	统计模块
appraisal	鉴定处置模块
use	利用模块, 包括下载、打印、导出、收藏夹、我的搜索条件
search	查询模块
reorganize	整理模块

4.5.2 文件的存放路径

文件路径	说明
/js/common	放置公共的 js 文件, 包括 ext、jquery
/js/模块名	放置各模块的 js 文件
/images/common	放置公共的图片资源
/images/模块名	放置各模块的图片资源
/jsp/common	放置公共模块的 jsp 文件

/jsp/模块名	放置各模块的 jsp 文件
----------	---------------

模块名参考包名定义

4.5.3 权限设计

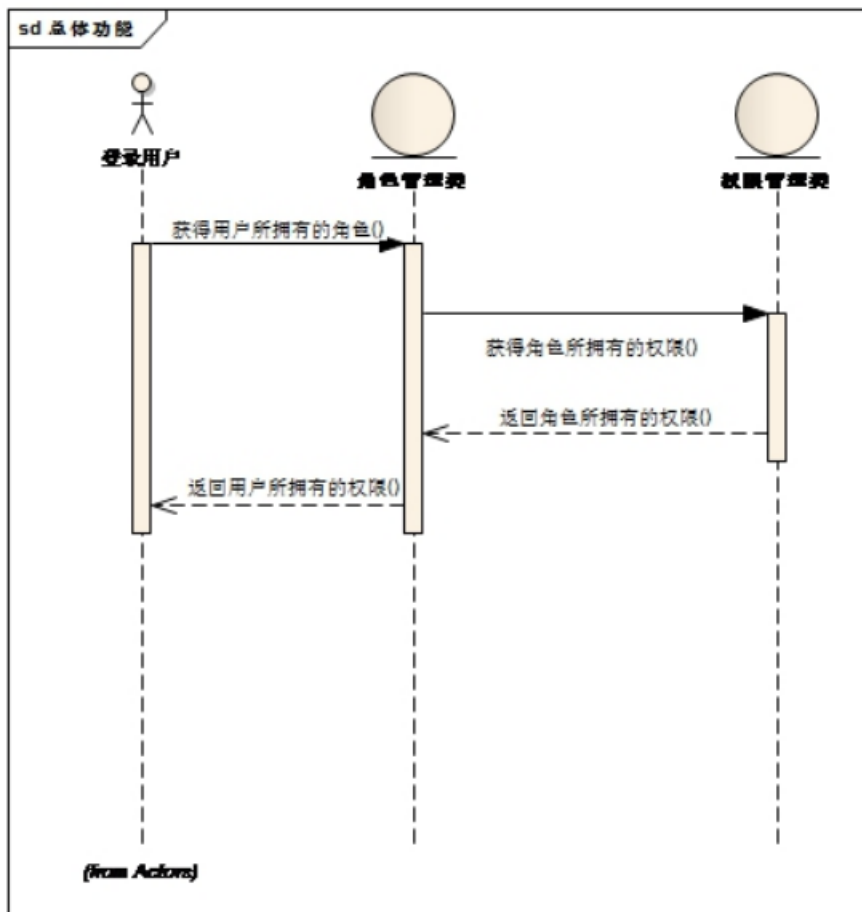
系统权限分模块级权限、功能级权限和数据级权限。

4.5.3.1 模块、功能权限设计

系统的模块级权限和功能级权限在用户管理的权限管理设定。

主要的模块有：文件管理配置、日志管理、用户管理、电子文件捕获、电子文件整理、电子文件统计。

主要的功能权限有：文件查看、文件导出、调单审核、文件审核。



权限获得流程如下：

4.5.3.2 数据权限设计

数据权限主要包括资产库的权限、文件记录的权限，组件的权限、业务行为的权限。

还有一类权限是文件的下载权限，组件下载权限通过组件的稿本默认权限和授权组合获得。

资产库的权限与文件记录的权限采用 ACL 授权的方式。

对于组件、业务行为的授权方式可能与资产有所不同，从业务角度来说，业务行为库根据行为授权而组件会根据稿本来授权，也就是说在查询模块要查看组件或业务行为库的时候，需要显示具有查看某些稿本的组件和某些行为的业务行为记录，组件的元数据是没有权限的，只有实体文件控制权限，实体文件的权限通过稿本的默认权限与借阅时授权获得。

稿本和行为的授权采用 ACL 授权的方式，在查询显示组件和业务行为库的时候需要作特殊处理。稿本和行为采用资产库来存储，自然稿本和行为的授权与资产的授权一样，参考资产的授权。

ACL 授权方式说明如下

4.5.3.2.1 特权

特权完全独立，权限的类型，主要有读、写等特权，这个版本只实现读的特权。写的权限在文件管理里用到，只需要有文件管理权限的角色对元数据。

4.5.3.2.2 授权种类

R 角色 G 用户组 U 用户 L 级别这个版本只做对角色的授权

4.5.3.2.3 权限继承方式

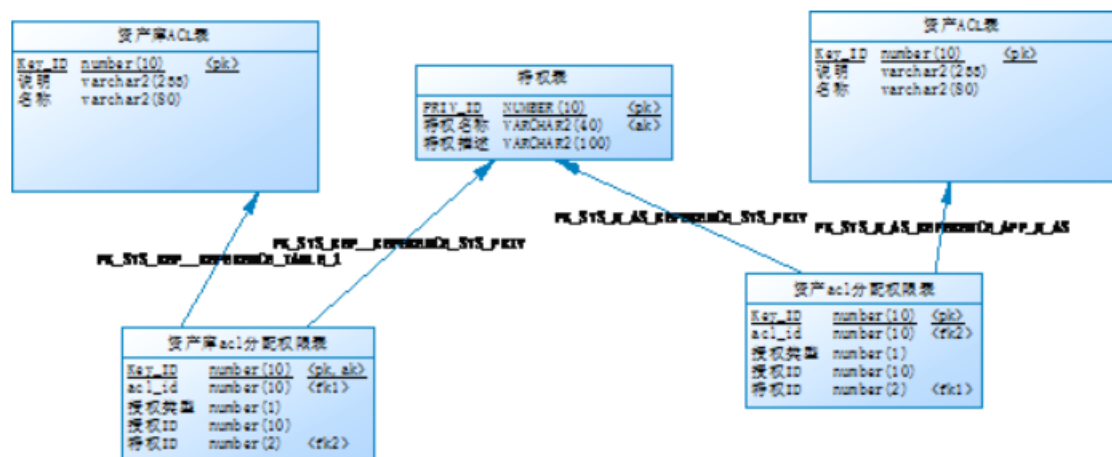
资产的权限自动继承资产库的权限。**权限的实现方式**

在每个授权对象上加一个 ACL_ID 的字段，授权的时候直接修改这个字段即可

登录用户自动获得所拥有的 ACL_ID 列表，通过 ACL_ID 列表可以获得相应的权限。

查询资产时只需要在查询语句的条件结尾加上 ACL_ID IN (ACL_ID 列表) 就可以实现权限的过滤，同理，全文检索也可以用同样的方式实现，权限的查询语种为 ACL_ID IN (ACL_ID1 or ACL_ID2...)。

注意：如果极端情况下，有很多的 ACL_ID，对性能产生影响，暂时不作考虑。



ACL 的数据模型如下图:

4.5.3.2.4数据存储服务

(一)资产 ACL 表 APP_ACL

名称	代码	注释	默认值	数据类型	长度	主键	外键
ACL_ID 主键	ACL_ID			NUMBER(10)	10	是	
Name	ACL 名称			NUMBER(10)	10		是

(二) 资产库 ACL 授权表 APP_ACL_AUTH

名称	代码	注释	默认值	数据类型	长度	主键	外键
KEY_ID 主键	KEY_ID			NUMBER(10)	10	是	
ACL_ID	ACL_ID			NUMBER(10)	10		是
授权类型	AUTH_TYPE	授权类型 R 角色 G 用 户组 U 用 户 L 级别		Varchar(1)			
授权 ID	AUTH_ID			NUMBER(10)	10		是
特权 ID	PRIV_ID			NUMBER(10)	10		是

(三) 特权表(sys_privilege)

名称	代码	注释	默认值	长度	主键	外键
id	priv_id	Id=1 代表 只读	主键		是	

名字	priv_name	非空唯一				
描述	priv_description					

priv_id	priv_name	priv_description
1	read	read
1	download	download

4.5.4跨库查询实现

跨库查询的实现思路，配置系统的联合文件库，配置同步的字段，通过触发器同步数据，其它的查询方式与普通资产库类似，不在整理里面出现跨库的文件库。

4.5.5基于 XML 的树型导航模块菜单设计

4.5.5.1树型菜单 XML 定义

菜单的XML表示如下：

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE ROOTSYSTEM "ROOT.dtd">
<模块配置>
<模块 priv="1" title="我的工作平台" icon="system">
  < 链 接 模 块  priv="2" title=" 我 的 工 作 任 务
  "target="rightFrame" href="a.html" icon="system">
    < 链 接 模 块  priv="2" title=" 我 的 借 阅 篮
    "target="rightFrame" href="a.html" icon="system"/>
    < 链 接 模 块  priv="2" title=" 我 的 调 阅 单
    "target="rightFrame" href="a.html" icon="system"/>
    < 链 接 模 块  priv="2" title=" 需 要 我 审 核 的 调 阅 单
    "target="rightFrame" href="a.html" icon="system"/>
    < 链 接 模 块  priv="2" title=" 我 的 借 阅 单
```

```

"taget="rightFrame"href="a.html"icon="system"/>
    < 链 接 模 块  priv="2"title=" 接 收 文 件
"taget="rightFrame"href="a.html"icon="system">
        < 链 接 模 块  priv="2"title=" 确 定 类 型 接 收
"taget="rightFrame"href="a.html"icon="system"/>
        < 链 接 模 块  priv="2"title=" 未 确 定 类 型 接 收
"taget="rightFrame"href="a.html"icon="system"/>
    </链接模块>
    < 链 接 模 块  priv="2"title=" 需 要 我 审 核 的 电 子 文 件
"taget="rightFrame"href="a.html"icon="system"/>
    </链接模块>
    < 链 接 模 块  priv="2"title=" 我 的 收 藏 夹
"taget="rightFrame"href="a.html"icon="system">
        <内嵌模块src="a.jsp"taget="rightFrame"href="a.html"/>
    </链接模块>
    < 链 接 模 块  priv="2"title=" 我 的 搜 索 条 件
"taget="rightFrame"href="a.html"icon="system">
    </链接模块>
</模块>
<模块priv="2"title="检索利用"icon="system">
    <内嵌模块src="a.jsp"taget="rightFrame"href="a.html"/>
</模块>
<模块priv="3"title="文件捕获"icon="system">
    < 链 接 模 块  priv="2"title=" 确 定 类 型 接 收
"taget="rightFrame"href="a.html"icon="system"/>
    < 链 接 模 块  priv="2"title=" 未 确 定 类 型 接 收
"taget="rightFrame"href="a.html"icon="system"/>
    < 链 接 模 块  priv="2"title=" 待 审 核 文 件
"taget="rightFrame"href="a.html"icon="system"/>
    < 链 接 模 块  priv="2"title=" 已 退 回 文 件

```

```

`target`=`rightFrame`href=`a.html`icon=`system`/>
    < 链 接 模 块 priv=`2`title=` 已 撤 回 文 件
`target`=`rightFrame`href=`a.html`icon=`system`/>
</模块>
<模块priv=`4`title=`文件整理`icon=`system`>
    <内嵌模块src=`a.jsp`target=`rightFrame`href=`a.html`/>
</模块>
<模块priv=`5`title=`文件统计`icon=`system`>
    <内嵌模块src=`a.jsp`target=`rightFrame`href=`a.html`/>
</模块>
<模块priv=`6`title=`文件鉴定处置`icon=`system`>
    <内嵌模块src=`a.jsp`target=`rightFrame`href=`a.html`/>
</模块>
<模块priv=`7`title=`文件管理配置`icon=`system`>
    < 链 接 模 块 priv=`2`title=` 文 件 分 类 配 置
`target`=`rightFrame`href=`a.html`icon=`system`/>
    < 链 接 模 块 priv=`2`title=` 数 据 字 典 配 置
`target`=`rightFrame`href=`a.html`icon=`system`/>
    < 链 接 模 块 priv=`2`title=` 元 数 据 方 案
`target`=`rightFrame`href=`a.html`icon=`system`/>
    < 链 接 模 块 priv=`2`title=` 流 程 配 置
`target`=`rightFrame`href=`a.html`icon=`system`/>
    < 链 接 模 块 priv=`2`title=` 存 储 配 置
`target`=`rightFrame`href=`a.html`icon=`system`/>
</模块>
<模块priv=`8`title=`安全管理`icon=`system`>
    < 链 接 模 块 priv=`2`title=` 用 户 管 理
`target`=`rightFrame`href=`a.html`icon=`system`/>
    < 链 接 模 块 priv=`2`title=` 日 志 管 理
`target`=`rightFrame`href=`a.html`icon=`system`/>

```

</模块>

</模块配置>

4.5.5.2XML 说明

说明如下:

这个文件放置在/WEB-INF/classes/pfile/conf/navigate.xml里。

将树形导航菜单中的节点分为两类,一类是叶结点,这类节点直接对应着功能页面选项。另一类是非叶节点,这类节点不对应具体功能选项,而只是作为多个相关的非叶节点或叶节点的双亲节点,发挥节点容器的作用。

《模块》节点生成导航菜单,生成效果如下图所示,树形导航菜单的由于受到性能的影响,这一层菜单只做一层。



《模块》有三个属性 priv, 权限值, 对应于权限管理的权限 ID 值, 0 表示不要权限, title, 节点显示名称 icon 节点样式,《模块》节点下可以有《内嵌模块》和《链接模块》。

《内嵌模块》直接引用别外一个 jsp 文件或 html 文件, 内嵌内容由引用的 jsp 或 html 控制

《链接模块》直接在模块下显示内容, 链接模块支持多层引用, 链接模块有五个属性: title, 其值表示节点在导航菜单上的显示文字; href, 其值用于表示要导航到的功能页面的 URL; target, 其值表示打开功能页面的框架名(一般设置为功能内容区所在的框架); icon, 其值表示节点文字前要显示的图标样式; priv, 该属性用于权限控制, 表示可访问该节点的用户所需权限 ID 值, 0 表示不要权限。链接模块下面也可以挂接内嵌模块节点。

《内嵌模块》节点有三个节点, src 指的是内容的内容的地址, 如果在模块节点下, 直接显示内嵌模块的内容, href, 其值用于表示要导航到的功能页面的 URL, target, 其值表

示打开功能页面的框架名（一般设置为功能内容区所在的框架）；

```
<?xmlversion="1.0"encoding="UTF-8"?>
```

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="http://www.saac.gov.cn/standards/ERM/encapsulation" targetNamespace="http://www.saac.gov.cn/standards/ERM/encapsulation" elementFormDefault="qualified">
```

```
<!-- 复杂元素的定义 -->
```

```
<xs:element name="模块配置">
```

```
<xs:complexType>
```

```
<xs:sequence minOccurs="0">
```

```
<xs:element ref="模块"/>
```

```
</xs:sequence>
```

```
</xs:complexType>
```

```
</xs:element>
```

```
<xs:element name="模块">
```

```
<xs:complexType>
```

```
<xs:sequence minOccurs="0">
```

```
<xs:element ref="链接模块"/>
```

```
</xs:sequence>
```

```
<xs:element ref="内嵌模块"/>
```

```
<xs:choice>
```

```
<xs:element ref="链接模块"/>
```

```
<xs:element ref="内嵌模块"/>
```

```
</xs:choice>
```

```
</xs:sequence>
```

```
</xs:complexType>
```

```
</xs:element>
```

```
<xs:element name="链接模块">
```

```
<xs:complexType>
```

```

        <xs:attributename="title" type="xs:string" use="required"/>
        <xs:attributename="url" type="xs:string" use="required"/>
        <xs:attributename="target" type="xs:string" use="required"/>
        <xs:attributename="icon" type="xs:string" use="required"/>
        <xs:attributename="priv" type="xs:string" use="required"/>

    <xs:sequence minOccurs="0">

        <xs:choice>
            <xs:elementref="链接模块"/>
            <xs:elementref="内嵌模块"/>
        </xs:choice>
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:elementname="内嵌模块">
    <xs:complexType>
        <xs:sequence>
            <xs:attributename="href" type="xs:string" use="required"/>
            <xs:attributename="src" type="xs:string" use="required"/>
            <xs:attributename="target" type="xs:string" use="required"/>

            <xs:attributename="icon" type="xs:string"/>

            <xs:attributename="priv" type="xs:string" use="required"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
</xs:schema>

```

4.5.6分类树状

4.5.6.1类目树状显示

用例名称:	类目树状
用例简述:	显示类目信息
执行者:	系统
操作界面:	
主要流程:	1) 列出资产库类别 2) 列出编目类别
替代流程:	
例外流程:	
业务规则:	1) 根据文件类别查询具体的资产库, 根据编码可以快速查询出文件信息。
算法:	

相关数据:	见附件数据字典中《资产库表》、《编目表》

4.5.7公共模块—文件信息

4.5.7.1文件列表显示

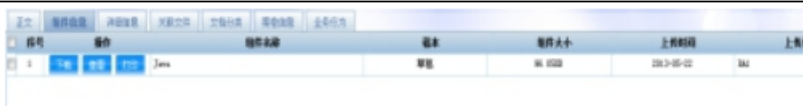
用例名称:	正文
用例简述:	列表文件的正文
执行者:	文档管理员、普通用户
操作界面:	
主要流程:	1) 在文件列表处点击文件标题 2) 出现文件正文文件;
替代流程:	2A) 没有正文, 查看是否有其它文件, 如果有显示其它文件
例外流程:	2B) 如果此文件没有附件, 则显示没有组件。
业务规则:	
算法:	1) 判断文件是否有组件, 如果有组件则判断文件是否有正文。 2) 如果文件有正文需要显示正文。
相关数据:	见附件数据字典中《文件表》、《组件表》

4.5.7.2组件列表

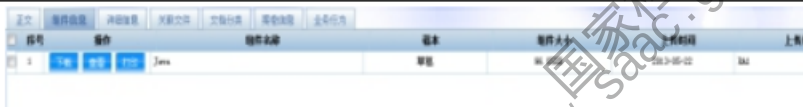
用例名称:	组件列表
用例简述:	列出文件下所有的组件
执行者:	文档管理员、普通用户
主要流程:	

操作界面:	
替代流程:	
例外流程:	
业务规则:	1)
算法:	此组件列表显示文件下的所有组件信息。
相关数据:	《组件表》/《接口表》

4.5.7.3组件下载

用例名称:	组件下载
用例简述:	选择组件，并对组件进行下载
执行者:	文档管理员、普通用户
主要流程:	
操作界面:	
替代流程:	
例外流程:	
业务规则:	1)
算法:	此组件列表显示文件下的所有组件信息。
相关数据:	《组件表》/《接口表》

4.5.7.4组件查看

用例名称:	组件查看
用例简述:	对文件下的组件进行查看
执行者:	文档管理员、普通用户
主要流程:	
操作界面:	

替代流程:	
例外流程:	
业务规则:	1)
算法:	此组件列表显示文件下的所有组件信息。
相关数据:	《组件表》/《接口表》

4.5.7.5组件打印

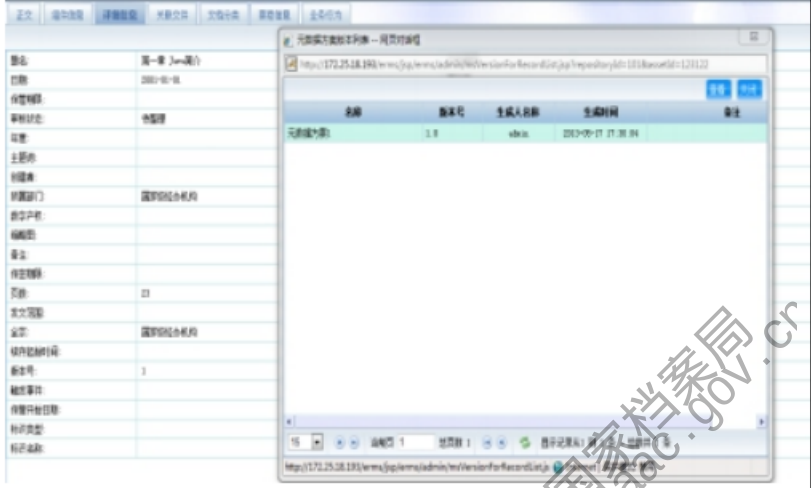
用例名称:	组件列表
用例简述:	打印文件下的组件
执行者:	文档管理员、普通用户
主要流程:	
操作界面:	
替代流程:	
例外流程:	
业务规则:	1)
算法:	此组件列表显示文件下的所有组件信息。
相关数据:	《组件表》/《接口表》

4.5.7.6文件详细信息显示

用例名称:	文件详细信息
用例简述:	文件详细信息
执行者:	文档管理员
相关用例:	

操作界面:	
主要流程:	根据查询文件详细信息。
替代流程:	
例外流程:	
前置条件:	
后置条件:	
业务规则:	
相关数据:	《文件表》

4.5.7.7元数据方案

用例名称:	元数据方案
用例简述:	元数据方案
执行者:	文档管理员
相关用例:	
操作界面:	

主要流程:	1) 点击元数据方案。 2) 在元数据方案中选择需要查看的元数据列表。 3) 在新的页面中显示此元数据方案的信息。
替代流程:	
例外流程:	
前置条件:	
后置条件:	
业务规则:	
相关数据:	《文件表》、《元数据方案表》

4.5.7.8显示关联文件

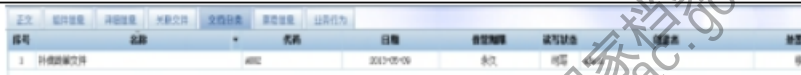
用例名称:	文件关联关系
用例简述:	文件关联关系
执行者:	文档管理员
相关用例:	
操作界面:	
主要流程:	显示关联文件
替代流程:	
例外流程:	
前置条件:	
后置条件:	
业务规则:	1) 所有的关联文件。
相关数据:	《文件表》、《文件关联表》

4.5.7.9显示业务行为

用例名称:	查询业务行为
用例简述:	查询此文件的业务行为。

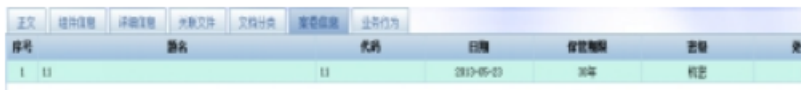
执行者:	文档管理员
相关用例:	
操作界面:	
主要流程:	1) 显示文件所有的业务行为
替代流程:	
例外流程:	
前置条件:	
后置条件:	
业务规则:	1) 查询此文件所有操作过的业务行为。
算法:	根据文件信息在业务行为表中查询此文档所有操作过的业务行为。
相关数据:	《业务行为表》

4.5.7.10显示文档分类

用例名称:	分类列表
用例简述:	查询此文件的分类信息
执行者:	文档管理员
相关用例:	
操作界面:	
主要流程:	1) 根据文件查询出文件所属的全部分类

替代流程:	
例外流程:	
前置条件:	需要选择文件信息
后置条件:	
业务规则:	根据文件显示此文件所有的分类信息。
相关数据:	《分类表》、《文件分类关系表》

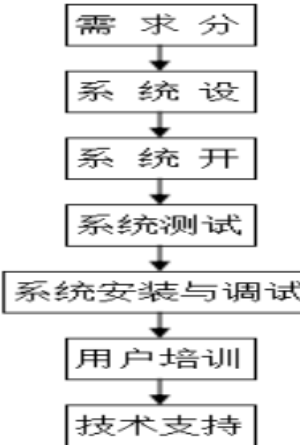
4.5.7.11显示所属案卷

用例名称:	案卷列表
用例简述:	根据文件、查询案卷信息。
执行者:	文档管理员
相关用例:	
操作界面:	
主要流程:	1) 根据文件信息、查询案卷列表。
替代流程:	
例外流程:	
前置条件:	
后置条件:	
业务规则:	1) 根据文件显示此文件所有属于的案卷 2)
相关数据:	《案卷表》、《案案关联表》

第 5 章宜昌市、县、乡、村文档一体化系统项目实施

5.1 软件项目实施过程总体控制要求

5.1.1 软件项目实施流程说明



5.1.1.1 实施流程汇总说明

5.1.2 软件项目职责分工说明

宜昌市、县、乡、村文档一体化系统在数据接口方面设计到很多的不同类型数据，在业务上设计到严谨的数据核销过程，多用户组、用户角色分配，为了支持本系统的复杂业务逻辑并保持本系统的灵活性、可扩充性，因此，需要各方面充分合作、互相协调、才能保证系统实施阶段顺利进行。

针对宜昌市、县、乡、村文档一体化系统要实现的功能，我们将建立如下的三层组织保证体系：

- 领导层
- 实施层
- 用户验收层

各组织层责任

项目领导小组的目的是和宜昌市、县、乡、村文档一体化系统项目小组共同建立对整个项目建设自始至终负责的、责权利相统一的管理，以形成一个由总体控制和有计划、有组织的控制网络系统，实现计划——实施——检查——调整——再实施的封闭循环。

领导层的具体职责如下：

- 明确成员各自分工与责任；
- 建立工程情况汇报制度；
- 决定工程重大原则问题；
- 协调各部门及相关人员关系。

对宜昌市、县、乡、村文档一体化系统数据库系统的实施，建立以项目中心领导小组为龙头，系统分析部、系统开发部、系统实施部为主体的，具有内部协调职能的直线式操作实施层，负责进度控制、成本控制、质量控制、供应控制。由于各实施环节均有项目质量控制小组评审，就有可能一次发现问题，并可及时落实到具体的负责部门，整个过程一气呵成，避免推诿扯皮现象和信息传递失误，从而提高系统建设效率。

作为组织保证系统的一个组成部分，还应建立线性责任系统，其目的是为了明确项目中各部门各成员在各项工作的地位与作用，使工作和责任一一对应起来，以保证组织运转更为有效。

实施层的具体职责如下：

- 建立项目实施小组，确定项目实施总负责人；
- 组织各专业技术队伍，明确技术责任与实施方案；
- 编制项目技术规范，计划进度，质量标准；
- 负责组织各专业技术人员优质按时完成项目任务。

用户验收层主要保证组织体系能有效地围绕总体控制计划运行，其人员由我们和**技术专家共同组成。

用户验收层的具体职责如下：

- 共同确定技术验收测试标准和验收测试人员；
- 组织现场测试环境和完成各项技术指标测试；
- 编写项目测试报告和项目验收报告；
- 组织并完成用户开通运行等各项工作。

5.1.3软件项目实施过程控制方法说明

5.1.3.1客户交流机制说明

为保证实现项目预期目标，项目实施过程中需要大量的客户交流工作，定期举行客户方

例会、向客户方提供阶段性报告、双方的主要联系方式是面谈、电话和 Email。

5.1.3.2 软件项目内部管理与进度控制方法说明

为了适应比较流行的 MVC 开发模式，保证开发小组成员顺利沟通，协同完成任务，我们拟采用较为普遍的软件工程项目管理方式。

在制定项目管理规范的过程中，我们将主要吸取 CMM 项目管理中的精华，紧密结合我们内部的实际和项目的实际情况，使规范具有较强的可操作性。

在项目实施各个过程中，为确保质量，必须明确各项活动的责任。即由谁负责、由谁辅助执行、工作应执行的流程、执行人员应具备的技能和培训经历等。责任越明确，可执行性就越好。同时要重视质量记录的产生和保存。在质量记录中，项目实施过程中产生的第一手原始资料是出现问题时查找原因的根据，也是系统可回溯性的保证。

设计是项目实施的初期步骤之一，它为整个项目的实施奠定基础，是保证项目质量的关键要素。在项目实施前，必须提交合格的设计文档。设计文档应充分考虑项目实施的实际情况，并从技术的角度确保正确可行。

5.1.3.3 软件项目发布方法说明

系统的安装与调试

本小组负责系统安装与调试工作，其中软件部分的安装是免费的。我们将派遣专业的技术支持人员进行现场的安装与调试，并向相关工作人员介绍安装与调试的要点，保证系统的使用人员掌握基本的系统安装的方法。安装与调试以系统能够正常稳定的工作为基本标准，兼顾用户的不同的使用要求。

服务器端相关软件系统可由硬件提供商负责，我们的技术支持人员也可以依据用户的要求对服务器端系统进行安装与调试。服务器端的数据库系统的建立、维护等工作将由我们的数据库设计工程师完成，保证数据库的稳定工作。

5.1.4 软件项目预期成果清单说明

5.1.4.1.1 文档成果清单

1. 《宜昌市、县、乡、村文档一体化系统需求规格说明书》

准确并详细说明用户的系统需求，开发人员以此为依据进行系统开发。也是向用户最终审核软件功能的标准。

2. 《宜昌市、县、乡、村文档一体化系统总体设计方案》

根据需求分析，设计出软件系统结构、所需的数据结构、系统内外部接口等，指导详细设计。

3. 《宜昌市、县、乡、村文档一体化系统详细设计说明书》

根据概要设计，对各模块功能进行细化，指导程序进行代码编写。

4. 《宜昌市、县、乡、村文档一体化系统测试方案》

进行软件的单元测试、集成测试和系统测试，完善软件的各方面功能。

5.2软件项目具体实施流程说明

5.2.1需求调研与分析阶段实施方案

5.2.1.1工作程序

1. 由项目实施小组组织需求方和开发方召开会议，需求方提出系统功能需求，开发方人员确定所需功能能否实现以及用何种技术实现，最终协商确定系统整体功能。
2. 项目实施小组提出需求分析说明书，并交由供需双方确认。

5.2.1.2工作内容与要求

- 1、描述用户需求，规划目标系统结构和主要功能，包括：数据接口，文件捕获登记、文件组织整理、文件鉴定处置、文件统计，文件检索利用、系统管理功能功能；
- 2、掌握用户特点，描述用户的角色、权限的划分，包括：系统管理员、文件管理员、一般用户；
- 3、了解设计与实现的约束条件，确定系统接口和运行环境，包括用户操作界面，网络传输与通信接口，硬件设备环境、软件配置环境；

5.2.1.3预期成果清单

《宜昌市、县、乡、村文档一体化系统需求规格说明书》

5.2.2系统设计阶段实施方案

5.2.2.1工作程序

开发组成员根据需求分析进行功能的分类汇总，确定系统架构及功能模块，提交《概要设计》和《详细设计》。

5.2.2.2 工作内容与要求

- 1、针对客户提出的各项需求，进行需求分类汇总，对功能、性能进行归并，为建立完整的系统结构、选择正确的实现策略奠定基础，从而形成相对稳定的功能模块划分。
- 2、针对需求的汇总分析和项目面临的各种约束和假定，进行重点难点问题分析，找出完成本项目所必须克服的重点和难点问题，一方面能够让技术人员理解相应功能模块的重要性，另一方面能够保证项目实施过程中所有人员都能够围绕重点和难点问题展开工作，使得项目研发的成果能够最大限度的满足客户要求。
- 3、系统模块划分，实现完整的功能模块划分，从而为技术人员的分工提供依据，这部分的内容要对上一节的内容进行更细致的分类，遵循技术形式进行划分。
- 4、系统接口策略，该项说明在系统实现过程中，需要与系统外部的硬件设备、网络结点、软件系统进行交互，并确定各类接口的类型、使用方法、所属的功能模块和本系统内各个功能模块之间的接口定义、相互依赖关系等。

5.2.2.3 预期成果清单

《宜昌市、县、乡、村文档一体化系统概要设计说明书》和《宜昌市、县、乡、村文档一体化系统详细设计说明书》

5.2.3 代码实现与测试调试阶段实施方案

5.2.3.1 工作程序

开发组成员根据《详细设计》确定各自分工，进行各个模块功能的具体开发实现工作，系统进入测试阶段后组织验收组编写测试报告。

5.2.3.2 工作内容与要求

项目开发组要保证开发进度的顺利进行，并将进度情况及时向用户汇报。

5.2.4 软件项目验收与交付阶段实施方案

5.2.4.1 工作程序

开发组提交项目完成报告，组织用户验收组进行项目验收。

5.2.4.2 工作内容与要求

在项目结束时，必须履行严格的验收和交接手续。终验和交接的目的是使用户对项目质

量和最终系统状态形成全面正确的认识,从而对系统集成工作满意和认可,同时向用户提交完整的文档,以便于用户日后对系统运行和维护。

在终验前,我们和客户共同组建终验协调小组,该小组应由各方领导小组委托各自的技术负责人组成,由用户方出任组长。该小组负责拟定终验测试方法、终验测试时间及周期、终验内容及终验进度安排,并且要对各厂商、集成商及最终用户的职责、义务与协调方法做出明确的规定。

终验应当在终验协调小组的领导下进行。终验测试过程要有详细的日志和质量记录。在终验测试完毕后,应综合提交终验测试数据和测试报告,经各方签字认可。

在终验过程中,项目质量监督小组应当有计划地进行质量审核和抽验工作,其审核报告应提交终验协调小组。

终验完成后,输出文件包括:终验测试方案、终验实施方案、质量记录、质量审核报告和终验测试报告。

终验结束后开始交接过程。交接是一个逐步使用户熟悉系统,进而能够掌握、管理、维护系统的过程。交接需要在终验协调小组的负责下有计划有步骤地进行。

交接包括技术资料交接和系统交接。系统交接一直延续到维护阶段(指合同中规定的无偿维护)。交接流程是:交接工作计划、所有文档清点和培训计划、系统交接与维护。

技术资料交接包括在实施过程中所产生的全部文件和记录。终验协调小组应督促各供应厂商提供相应的技术资料,并整理网络、服务器和数据库的所有配置参数及其它数据。

交接时至少提交如下资料:总体设计文档、项目实施设计、系统配置文档、测试报告、系统维护手册和系统操作手册(可由设备厂家提供)以及系统管理建议书。

在技术资料交接之后,便进入维护阶段。维护阶段的时间长短由终验协调组协商确定,且各方的任务要明确。维护阶段的主要任务是协同用户进一步熟悉和掌握整个系统,提高用户自身的系统管理能力。

5.2.4.3 人员分工与资源配置

由工作组完成。

5.2.5 客户服务与技术支持阶段实施方案

5.2.5.1 工作程序

确定系统交付使用后的客户服务与技术支持的内容。

5.2.5.2 工作内容与要求

5.2.6 系统的安装与调试

我们负责系统安装与调试工作，其中软件部分的安装是免费的。我们将派遣专业的技术支持人员进行现场的安装与调试，并向相关工作人员介绍安装与调试的要点，保证系统的使用人员掌握基本的系统安装的方法。安装与调试以系统能够正常稳定的工作为基本标准，兼顾用户的不同的使用要求。

服务器端相关软件系统可由硬件提供商负责，我们的技术支持人员也可以依据用户的要求对服务器端系统进行安装与调试。服务器端的数据库系统的建立、维护等工作将由我们的数据库设计工程师完成，保证数据库的稳定工作。

5.2.7 用户培训

培训的主要目的是为了使宜昌市、县、乡、村文档一体化系统的用户能更好的掌握新系统的安装、使用和管理方法，充分发挥系统的作用，使他们经过系统培训后，能够对系统进行日常的使用和管理工作。为了保证培训的质量，我们将把培训的内容做成多媒体教学光盘，辅助培训教学，并方便用户日后的使用。

培训针对不同岗位职责的具体业务的需求，在力求覆盖绝大多数的有各种工作人员的要求下，对领导、干部、技术主管等不同类别用户，设计相应的培训方案。通过对用户的归类、划分，我们可以尽可能的使培训落实到每一个用户的具体职责上去，保证良好的培训效果。我们将针对国家新农合电子文件管理系统人员的特点，准备两套培训内容：

初级培训，系统建设和操作维护。

高级培训，系统深层次应用开发。

操作维护培训和高级培训包括：

所提供软件的工作原理和技术性能、操作维护方法、安装调试、排除故障及软件结构、定制和升级等各个方面，系统管理和建设的软件系统的使用，以及系统进一步开发的全套方法和接口协议。

并提供全套培训教材和培训课程计划表。

对于参与项目的宜昌市、县、乡、村文档一体化系统人员我们将在项目的开发设计过程中进行有侧重培训，使他们充分了解系统设计的体系架构、开发规范和项目管理方法等。

我们将准备如下内容课程：系统操作、系统维护与运营和用户使用培训等课程，另外，

我们将提供在线式的帮助系统。

5.2.8 系统维护

系统维护是指系统在功能上进行修改，由于宜昌市、县、乡、村文档一体化系统业务需求的扩展性，在实际应用中，特别是在系统刚投入使用时，需经常改动程序功能。针对这种情况，我们的系统维护工作从以下几个方面入手，从而使系统应用达到预期目的。

- 在系统设计阶段应充分考虑系统的灵活性，将各个功能模块设计成可由系统维护或用户定制的，如：对各项固定参数的代码表的维护，通用的指标定制，通用定制查询条件，通用的报表设计等。这样系统的应用范围就会变得很广，通常只要数据库中存在的数据都可以由系统按照用户的需求进行组合输出。灵活的程序结构理想状态可以做到只要数据库不作改动，程序代码也无需改动。对于数据库，因在机构上予以优化，给予其较大的自由度，但数据库变动一般影响较大，过于灵活的设计会影响系统运行效率，因而数据库结构往往会随着需求的变化而调整。

对于某些情况，如：数据库结构的改动，仍需要对程序代码进行修改。对于该类维护，在系统试运行阶段开发商应在系统运行地实施免费维护，进一步使需求达到用户要求，最大限度地做到让用户满意，系统试运行阶段一般为系统验收通过后二个月。系统试运行期过后，开发商将与用户另行签订维护合同，以下几点均是对系统试运行期后维护工作而言。

- 业务框架的更改：当业务流程发生变化，并在系统中需要改动的，此类变动涉及面很大甚至要重新架构部分功能模块的，我们可以先进行系统升级具体费用可在后期合同中体现。

本情况的处理方法为：双方认真讨论新需求，结合已有系统的现状提出可行方案。之后，由开发商提交解决方案和时间进度表，共同讨论通过后，开发商进行系统分析和详细分析。将新系统升级完成后，在测试环境下运行，测试通过后将全面上线试运行。试运行无误后，提交相关文档。在此过程中除开发期均有系统分析员参与讨论或工程师现场维护直到能够正式运行。

- 单一模块的功能升级：随着业务需求的不断变化，将出现现在无法预测的变动，而本系统的相应模块未能适应该变动，例如：业务内容、操作方法、管理控制、界面风格等。此时需要代码升级，原则上提供免费升级，工作量超过 7 人日则采取先升级，日后在后

期合同中加以体现的方法运作。

本情况的处理方法为：由客户提出需求，双方就系统现状提出可行方案。之后，由开发商提交解决方案和时间进度表，共同讨论通过后，开发商进行系统分析和详细分析。将新系统升级完成后，在测试环境下运行，测试通过后将全面上线试运行。试运行无误后，提交相关文档。在此过程中除开发外，均有系统分析员参与讨论或工程师现场维护直到能够正式运行。

- 简单的功能升级：有些内容的变更不在系统自动升级的范畴以内，将需要工程师的介入。
本情况的处理方法为：此时工程师在现场明确国家新农合电子文件管理系统的需求后进行现场开发，开发完毕后在测试环境下运行，测试通过后将全面上线试运行。试运行无误后，2日内提交相关文档。工程师在现场进行维护直到新需求能正常使用。
- 新增功能：随着新的理念、管理方法、分析手段和功能应用的出现，会有新的功能模块的需求出现。

本情况的处理方法为：在双方认真讨论新需求后，由开发商提交解决方案和时间进度表，共同讨论通过后，开发商进行系统分析和详细分析。将新系统升级完成后，在测试环境下运行，测试通过后将全面上线试运行。试运行无误后，提交相关文档。在此过程中除开发期均有系统分析员参与讨论或工程师现场维护直到能够正式运行。

- 数据库运行环境的变更。包括：数据库服务器的更换、数据库数据阶段性备份和整理、数据库管理系统的升级等。

本情况的处理方法为：由客户提出需求，双方就新需求提出可行方案。之后，由开发商提交解决方案和时间进度表，共同讨论通过后，开发商在测试环境下实施，测试通过后转移到新环境下试运行。试运行无误后，提交相关文档。在此过程中除文档编写期均有系统分析员参与讨论或工程师现场维护直到能够正式运行。

- 数据库结构更新。包括：增加、删除系统表和数据表，改变系统表、数据表结构，建立备份数据表等。由于业务、功能和流程上的需要，表结构的更新在所难免，本变动一般是围绕着业务进行的。

本情况的处理方法为：在业务处理过程中，讨论此处的更新，处理过程同业务流程，工作量记入业务处理中。

- 系统自适应升级的维护：当新的需求满足系统自动升级的条件时，为确保升级能够正确，并保证系统的稳定，我们有工程师在现场指导维护，并处理突发事件。

本情况的处理方法为：开发商派工程师到现场监控升级过程，直到系统能在升级后的条

件下正常运行。

5.2.9技术支持和维护的方式

以下为我们提供技术后援支持服务的主要方式:

➤ 现场技术支持服务:

我们将安排经验丰富的工作人员到现场进行维护。为了让用户的问题得到尽快的处理,我们将为所有的现场维护技术支持配备手机、呼机。

为了有效地进行系统软件的维护,我们事先就开始做组织工作,建立专为该项目成立的维护的机构,声明提出维护申请报告的过程及评价的过程,为每一个维护申请规定标准的处理步骤,还必须建立维护活动的登记制度以及规定评价和评审的标准。

➤ 电话咨询:

我们设立热线服务台,提供 7*8 小时(即每周 7 天,每天 8 小时)电话技术咨询服务。

➤ 网上技术支持:

通过电子邮件、BBS 等方式提供 7X24 小时的服务。我们并把一些技术维护文档上传到国家新农合电子文件管理系统网上,系统用户可以根据自己的需要在网上下载后使用。

➤ 技术资料:

在系统正常运行后为用户提供一整套所安装软件的维护手册,其中包括详细的设计手册、安装手册、使用手册。如果以后软件修改、补充或升级时,我们都会更新升级维护手册;而且我们还会定期为用户递送一些资料及产品公布信息,同时以上资料也将在网上发布并及时更新。

➤ 响应时间:

对工程任何层次的维护要求,我们都将建立 4 小时内作出响应的机制,并且根据站点的重要性及其复杂程度的不同,我们将相应缩短维护的响应时间。

第 6 章 电子文件与电子档案全程管理的标准体系

6.1建设标准规范体系

标准规范体系是由一定范围内的具有内在联系的标准组成的有机整体,它包括现有的、正在制定的和应着手制定的各类标准,是促进一定范围内的标准组成趋向科学化和合理化的

手段，通常用标准体系框架和明细表的方式来表达。

标准规范的建设是一项复杂而艰巨的任务，工作量大，且需要协调的方面很多，因此在建设的过程中要必须遵循以下原则：

- 统一标准，保障安全

统一标准，保障安全是本试点项目建设必须坚持的重要原则之一。

- 切实可行、准确实用

标准和规范必须根据实际情况而制订和修改，这样才能使标准符合实际，标准的制订和修订要求准确使实用，使执行者易于理解和执行，具有较强的可操作性。

- 遵循电子政务和电子文件管理的国家标准、行业标准和地方标准

标准和规范的制订应遵照、继承和贯彻国家标准、行业标准和地方标准，避免重复建设。

依据国家标准化工作相关文件，结合《电子文件管理标准体系框架》和宜昌市电子文件相关标准，以及本课题的实际，按照“统筹规划、国家主导、统一标准、联合建设、互联互通、资源共享”的实施原则，初步搭建宜昌市电子文件标准规范体系。包括以下内容：

- 《归档文件整理规则》
- 《机关文件材料归档范围和文书档案保管期限规定》
- 《会计档案案卷格式》
- 《会计档案管理办法》
- 《科技档案案卷构成的一般要求》
- 《企业档案工作规范》
- 《印章档案整理规则》
- 《照片档案管理规范》
- 《湖北省机关档案工作业务建设规范》
- 《宜昌市档案局关于实施相关档案工作业务新规范的通知》
- 《宜昌市档案目录数据采集与报送工作的基本要求》
- 《宜昌市纸质文书档案数字化操作规程》
- 《宜昌市通用电子文件元数据方案草案》
- 《宜昌市电子文件数据存储结构要求草案》
- 《宜昌市电子文件封装要求草案》
- 《宜昌市县乡村文档一体化系统用户编码方案》
- 《宜昌市县乡村文档一体化管理系统著录规则》

- 《宜昌市纸质档案批量命名系统》

6.2 关键环节的标准化工作

本项目是通用电子文件与电子档案全程管理的研究,为此在全程一体化的关键应用环节,进行了以下七项标准的编制工作。

- 《宜昌市纸质文书档案数字化操作规程》
- 《宜昌市通用电子文件元数据方案草案》
- 《宜昌市电子文件数据存储结构要求草案》
- 《宜昌市电子文件封装要求草案》
- 《宜昌市县乡村文档一体化系统用户编码方案》
- 《宜昌市县乡村文档一体化管理系统著录规则》
- 《宜昌市纸质档案批量命名系统》

第 7 章 电子文件与电子档案全程管理应用实践

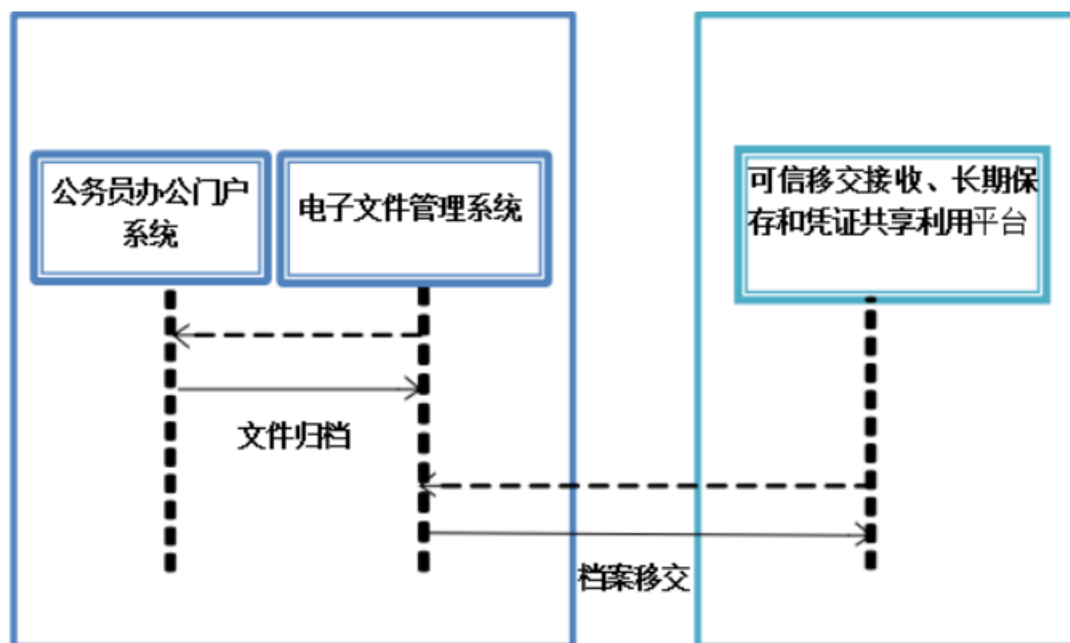
7.1 技术策略

基于电子文件全生命周期管理的基本定位,电子文件生命周期的全过程必需贯穿于系统业务流程的始终。

完成电子文件全程管理的应用建设,涉及到通用电子文件多个相关业务应用系统的整合,包括业务数据、业务流程和业务功能等多维化的整合、集成和改造工作。通过对宜昌市电子文件与电子档案实际需求调研,梳理出满足各系统业务功能的要求,并对各应用系统的业务逻辑关系以及所承载的电子文件与电子档案业务信息数据的形态进行充分研究分析,结合宜昌市电子文件与电子档案在电子政务中的应用特点,形成了安全有效的技术路线和设计方案。

从电子文件与电子档案管理业务全生命周期过程的纬度,实现各业务环节所涵盖的业务应用系统、业务管理流程以及业务管理数据流向的全程贯通。

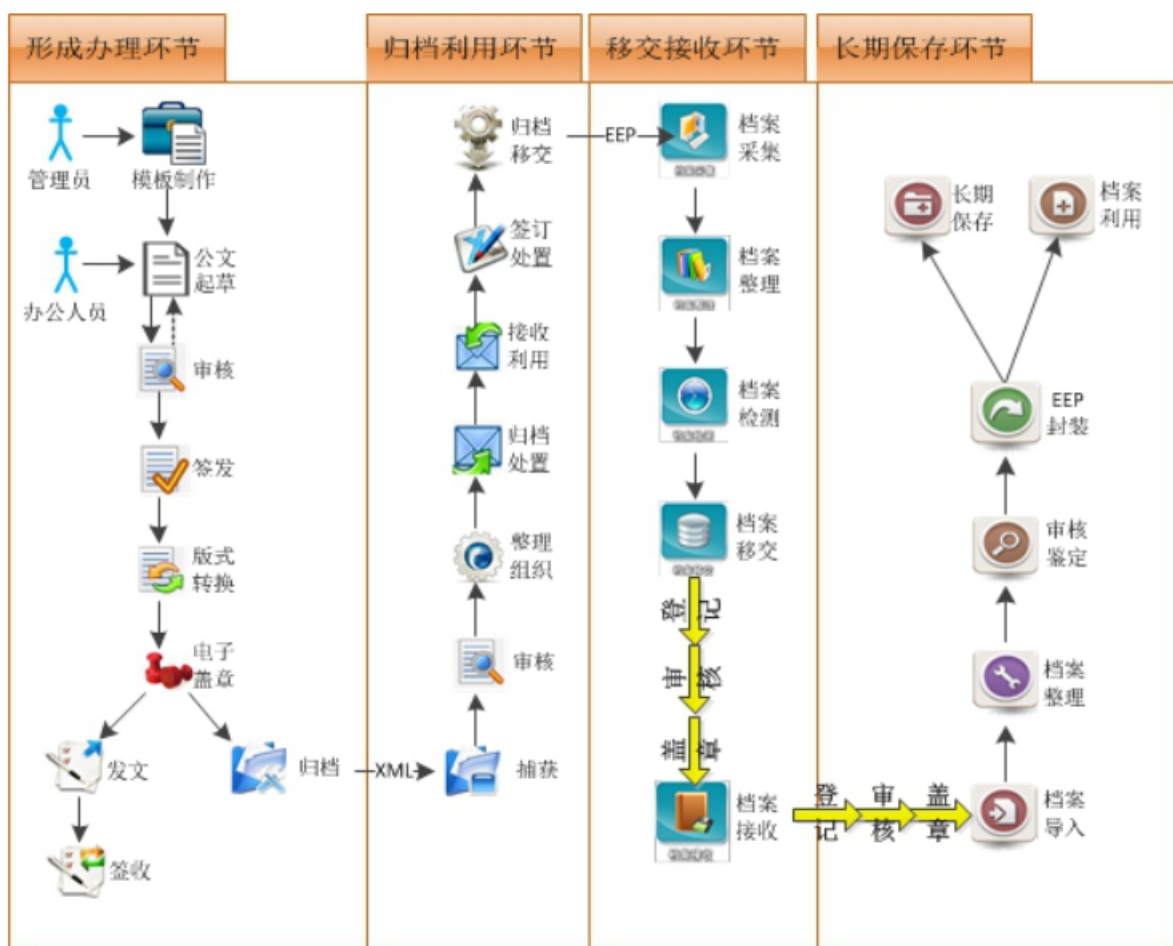
7.2 技术架构研究



实现电子文件全程管理依赖于电子文件业务处置体系的应用管理系统, 宜昌市电子文件与电子档案的全程管理, 是从电子文件形成办理业务的公务员办公门户系统出发, 通过归档和共享的电子文件管理系统, 终了至可信移交接收、长期保存和凭证共享利用平台。

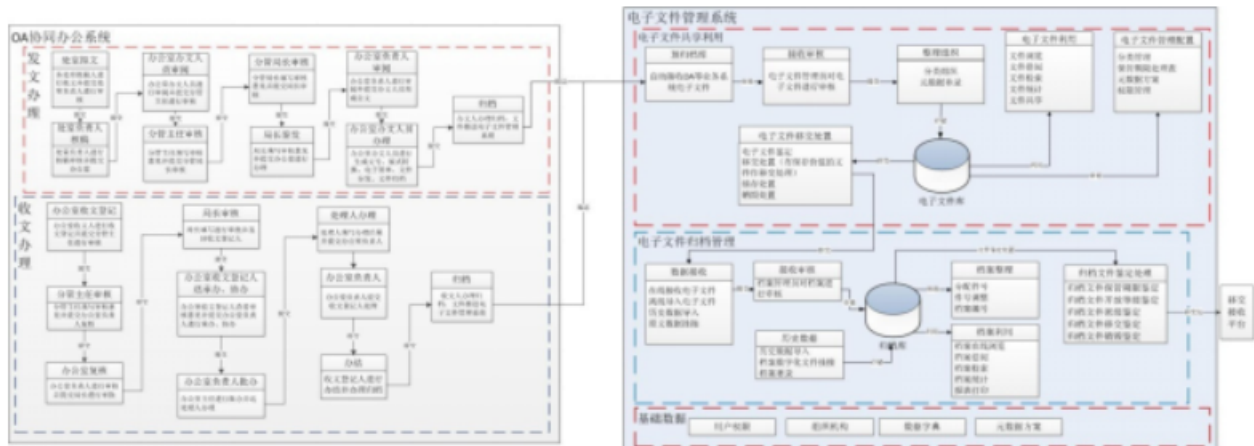
7.3 业务流及数据流研究

实现电子文件与电子档案全程管理的业务流程集成的关键是在于切实把握宜昌市电子文件与电子档案全程管理的业务特点, 在电子文件全程管理的概念下, 对业务流程的进行再造, 实现电子文件业务流和数据流的全程一体化贯通。



电子文件与电子档案全程管理业务跨四种业务应用。在电子文件形成办理环节，通过公务员办公门户，实现了电子文件形成办理的规范化、电子化；归档后在归档利用环节，通过实施电子文件管理系统实现电子文件的共享利用和电子文件的归档管理；在移交接收环节，通过实施市县两级集中式的移交接收系统，实现全市电子档案移交接收管理；最终在长期保存环节，通过实施电子档案长期保存系统，实现电子档案的长期保存和安全存储。

在电子文件归档管理环节采用《电子文件管理系统通用功能要求》的对标应用系统的实施，并且针对实际业务的应用需求，增加了档案处室的管理功能，实现电子文件归档及共享利用的统一管理，同时实现业务流和数据流的应用集成，如图所示。



7.4业务流及数据流研究

7.5关键技术难点

实现电子文件与电子档案的全程管理应用，采用国有自主知识产权的关键技术，以版式文件技术为核心，形成了电子公章技术、数字签名技术和电子文件封装技术相互支撑的关键技术链。

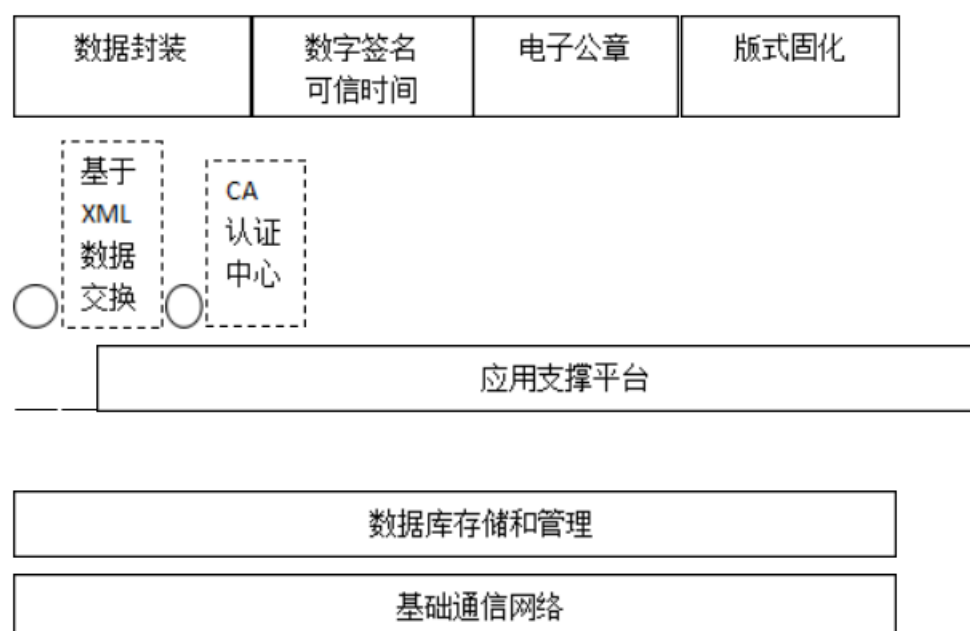
CEBX 版式技术贯穿于电子文件全程管理各业务系统始终，为电子文件真实、完整、可靠和可用提供了技术支撑。

在电子文件成文环节，使用合法的数字证书、数字签名和加密手段制作电子公章，并结合密钥发放。采用 CEBX 版式文件的电子公文加盖电子公章，使得电子公文与电子公章不可分割、不可篡改。

当电子文件需要进行长期保存时，通过对电子档案进行 EEP 数据包的封装，将实体电子档案的 CEBX 格式文件和各种元数据文件形成完整的数据包，保证电子文件的真实、完整、安全和可靠。

上述自主可控的关键技术所形成的关键技术链，成就了电子文件全程管理的技术实现，是电子文件全程管理必备的组件工具。

电子文件全过程管理应用界面



第 8 章 课题研究成果

8.1 解决的难点问题

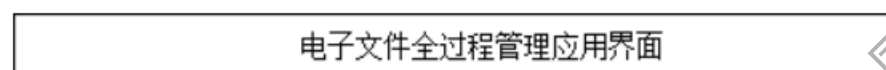
实现电子文件与电子档案的全程管理应用，采用国有自主知识产权的关键技术，以版式文件技术为核心，形成了电子公章技术、数字签名技术和电子文件封装技术相互支撑的关键技术链。

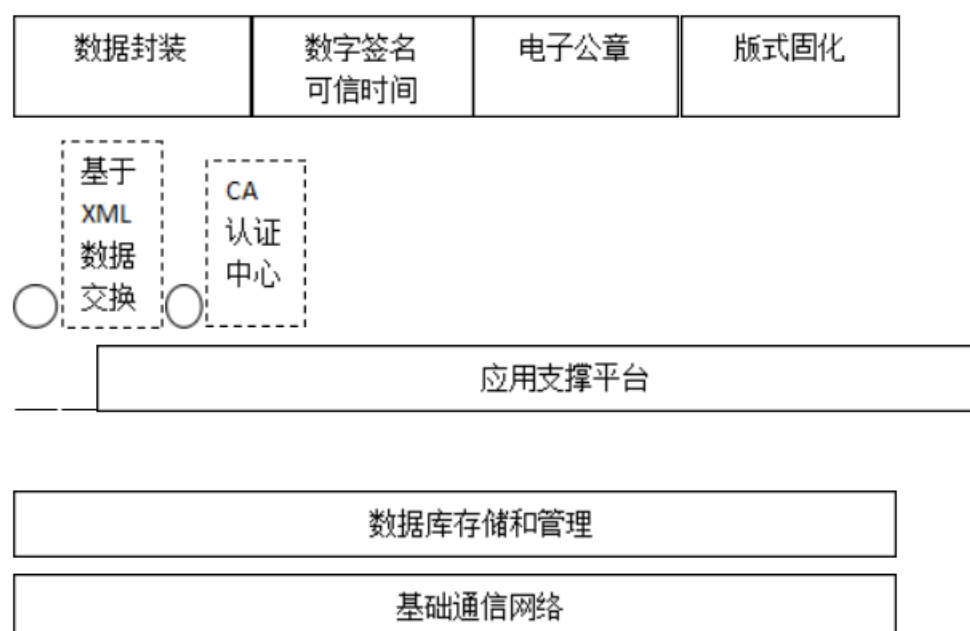
CEBX 版式技术贯穿于电子文件全程管理各业务系统始终，为电子文件真实、完整、可靠和可用提供了技术支撑。

在电子文件成文环节，使用合法的数字证书、数字签名和加密手段制作电子公章，并结合密钥发放。采用 CEBX 版式文件的电子公文加盖电子公章，使得电子公文与电子公章不可分割、不可篡改。

当电子文件需要进行长期保存时，通过对电子档案进行 EEP 数据包的封装，将实体电子档案的 CEBX 格式文件和各种元数据文件形成完整的数据包，保证电子文件的真实、完整、安全和可靠。

上述自主可控的关键技术所形成的关键技术链，成就了电子文件全程管理的技术实现，是电子文件全程管理必备的组件工具。





8.2 创新点

本课题研究是实现电子文件与电子档案的全程管理，其价值取向是通过实现全程管理，提高通用电子文件与电子档案业务的效能。强调电子文件的作用与价值统一，管理目的与管理手段统一，管理过程与管理结果统一。

1、实现了全程数据的规范化管理

在电子文件与电子档案全程管理中，数据的真实、完整及其显示的规范是应用中的基本，通过标准化的手段，建立电子文件的规范生成、实现生成后的电子文件在全程管理过程中的版式文件“一版到底”的数据格式理念，完成电子文件和纸质文件的一致性、打印和印刷的一致性，通过屏幕自适应的功能，完成各种客户端显示一致性。

2、实现了全程数据的安全化管理

保证电子文件的原始性和真实性是解决电子文件与电子档案凭证价值的核心问题，是电子文件作为电子政务运行工具的重要条件和根本基础。通过防扩散、防泄漏、防伪造等保真防护措施，通过数字签名、电子签章等权限行为控制措施，以及对行为可追踪可鉴别等系统运行中的行为管理措施，保障电子文件的数据安全。

3、全程管理效能价值

在项目建设中通过统一了标准规范，形成了电子文件拟制、批办、盖章成文、收发文、可传输、可交换、可管理、可利用、可共享、可长期保存的安全应用的集成环境，实现了电

子文件全程贯通的业务目标，解决了电子文件数据的真实性、完整性、可靠性和可用性问题，充分体现了全程管理的效能价值。

4、跨区域多级全程管理吻合业务

实现电子文件与电子档案业务应用系统全程贯通后，其应用功能和业务流程与以往的现实工作相吻合。

电子文件与电子档案全程管理对重构完整的电子文件与电子档案[价值链](#)的要求越来越高，但电子政务信息化建设成功的关键，就是在再造业务流程后、不改变业务管理职能，以利于信息系统得到推广和应用。

特别是本课题是基于全市的跨区域、跨机构、跨应用、多级业务管理，实现公务员门户、电子文件管理系统、电子档案移交接收系统，电子档案长期保存系统的数据集成、业务流程集成，并且在电子文件管理系统中完成电子文件的共享利用和归档管理的两种业务功能。通过规范化、优化业务流程，再造全程贯通的电子文件与电子档案数据管理，在吻合各部门以往日常工作职能的前提下实现了电子文件与电子档案的全程管理。

5、区域集中式管理应用的权限模型创新设计

- 多维的应用权限控制模型设计。依据市、县、乡、村四级进行模型设计，实现分级管理、分组授权管理。应用权限包括“模块级权限”、“功能级权限”和“数据级权限”。
- 多级组织机构权限模型设计。系统组织机构范围是全市的电子政务网覆盖单位，每个覆盖单位能够设置一个或多个全宗。
- 安全控制管理。按照“三权分立，相互制约”的思想，在集中式系统中设计了系统管理员、安全管理员和审计管理员，系统管理员只负责系统元数据用户、单位、全宗的日常维护，用户的授权操作由安全管理员完成，审计管理员对系统管理员和安全管理员的操作进行监督管理。
- 多种角色管理。按照全市的业务需求，设计了 10 余种用户角色，保障系统的细颗粒度的权限划分。

6、创建区域机构身份标识编码体系

采用 25 位制编码机制，编制全市机构的身份标识，为所有单位建立唯一身份标识，以有效解决暂未纳入进馆范围全宗的数据管理问题，从根本上消除数据混乱现象的发生。

7、实现四级纵横交错的远程利用机制

以县市区行政区为单元，在区域内，档案室之间可横向远程利用，档案室与档案馆之间可纵向远程利用。在区域间，档案馆之间可横向、纵向远程利用。借助这个纵横交错的四级利用平台，任何一个公民，都能够利用到全市区域内的电子文件和电子档案信息。