## 系统总体设计

根据国家和本市电子政务建设的指导思想，结合公证行业的具体特点，设计过程坚持了开放性、规范性、可靠性、安全性、规范性、可维护性、兼容性、经济性，易用性，先进性与成熟性相结合的原则。本章对系统的体系架构、功能结构、网络拓扑、接口和系统框架的设计方案进行了论述，过程中基于业务需求、网络环境、技术平台、以及相关标准和规范，提出了系统的总体设计方案，并论述了方案的可行性和有效性。

### 体系架构

通过系统建设需求和应用模式的分析，根据系统体系架构的不同特点及性质，在遵循统一规划、四网互动，同时考虑应用政府已有资源的情况下，为了从更宏观的角度去认识系统的设计思想，解决软件系统的复杂度，对总体架构进行了分层设计。系统采用了四层体系架构，分别是展现层、应用层、支撑层和数据层，同时有坚实的信息安全保障体系，统一明确的技术标准和规范体系，如所示。



**图 15 系统体系架构图**

系统的四层体系架构自上而下分别是：

（1）展现层

展现层包含了应用程序中需要和用户交互的组件。例如：Web页面，Rich-client表单，用户交互处理组件等。AndroMDA当前提供了两种技术来构建Web基础的展现层：Struts和JSF，该层通过值对象与应用层进行数据交互。课题应用了Struts技术解决了用户的个性化要求，比如：表单验证、列表的排序和样式表的管理等。

（2）应用层

应用层也叫业务层，是基于支撑层定制的满足业务需求的各种应用系统，主要有公证管理系统、帮教安置系统、社区矫正系统、司法鉴定系统、数据分析服务系统，此外还包括业务系统间的内部接口，以及与外部系统间的接口。通过接口，可以实现业务内部管理部门之间、以及本市各相关业务部门（公安、民政、银行等）之间的信息交换和共享。该层封装了应用业务功能的核心，业务组件只提供服务接口，而隐藏了复杂的业务逻辑。业务层主要由AndroMDA生成的服务组成，使用Spring框架来配置服务。AndroMDA也能生成Web Services服务或EJB，还能为工作流引擎生成业务过程和工作流。该层通过实体类或值对象类与数据访问层进行数据交互。

（3）支撑层

支撑层处于操作系统与用户的业务系统之间，为业务系统提供运行和开发环境，帮助用户灵活、高效地开发和集成复杂的应用软件，为各个业务系统间的互联、互通、互享、互动提供支持，提供异构数据及异构软硬件平台之间的协同机制，它不仅是业务应用系统的支撑，而且是实现业务系统之间信息共享和互连互通的通道。课题中的工作流、报表和数据交换部分分别应用了西安协同的工作流产品、润乾报表工具和东方通的数据交换软件。

（4）数据层

数据层是整个系统的核心，分为数据访问层和数据存储层。数据访问层提供了一种简单的访问和操作数据的应用程序接口，这一层抽象底层数据访问技术，因而允许业务层重点关注业务逻辑。AndroMDA已流行的对象关系映射（object relational mapping，ORM）工具Hibernate为基础生成访问层。应用系统将数据保存在一个或多个存储结构中，数据库和文件系统是两中常用的数据存储结构。由于AndroMDA使用Hibernate来生成对数据的访问，用户可以使用Hibernate支持的任何数据库，常用的如：MySQL、Oracle、SQL Server、DB2等，这里使用的是Oracle10g实现对系统产生的各类信息和数据以及其他各类扫描件、音频、视频等非结构化信息的集中存储与管理，公证管理系统的数据主要分基础库、业务库、共享库和在线库。该层中的Hibernate利用与数据库的映射关系对数据库和记录进行更新和浏览操作。

### 功能结构

完成了系统体系架构设计，解决了系统中的层次结构，接下来将对系统的功能结构进行设计，功能结构划分主要是根据系统功能定位和方便易用的原则进行设计。系统主要完成三方面的内容，一是完成公证处的办证和日常办公工作，另一个就是完成行政许可和业务监管，第三个就是公众的咨询、投诉和查询服务。因此，系统在总体模块划分上也分为三个子系统：公证处综合管理子系统、公证监管业务子系统和公证在线服务子系统，并根据不同的工作内容侧重及用户的方便易用，为每个子系统设置了相应的功能模块。如所示。



**图 16 公证管理系统功能结构图**

公证处综合管理子系统为全市所有公证机构提供一个办公平台，支撑日常办证管理和办证辅助管理，同时实现市局公管处、区县局、协会、各公证处之间的信息交流、数据共享和通知通告等，方便公证处日常办公和管理。其功能模块划分如所示。



**图 17 公证处综合管理子系统功能结构图**

市、区（县）两级司法行政机关和公证协会协同对公证业务进行监督管理。其中公证协会负责：投诉管理、公证协会业务管理、查询统计；市局和区县司法局主要负责：公证机构管理、公证员管理、奖惩情况管理、投诉管理、查询统计等业务。其功能模块划分如所示。



**图 18 公证监管业务子系统功能结构图**

社会公众通过公证在线服务子系统进行公证业务方面的咨询，公证协会等工作人员再给予问题的解答等，同时可以查看本市的任何一个公证处的公共信息，方便社会公众前去办理公证业务；会员可以登录在线服务，查询所办理的公证书的详细信息。其功能模块划分如所示。



**图 19 公证在线服务子系统功能结构图**

### 网络拓扑

网络拓扑结构是指用传输媒体互连各种设备的物理布局，就是用某种方式把网络中的服务器、防火墙、交换机和VPN等设备连接起来。根据应用系统的安全性要求和网络现状，各服务器部署在不同区域，如所示。



**图 20 网络拓扑结构图**

（1）内网区域：属于安全区域，为司法内部用户提供各种服务，WEB/应用服务器、数据库服务器和已有统一认证服务器部署在本区域。应用服务器采用双机备份的方式，利用windows2008实现双机集群，利用Apache实现负载均衡，提高系统的性能和降低系统故障率；数据库服务器也采用双机热备的方式，提高数据的安全性，通过磁盘阵列增大存储容量，解决系统中产生的大数据量问题。同时，内网与政法委专网通过网闸进行连接，实现安全的数据共享服务。

（2）政务专网区域：为区县司法局和其他委办局提供系统访问和数据共享服务，数据同步服务器用于配合网闸实现网闸内外的数据同步，从而实现业务上与公、检、法、监狱、劳教等部门的信息交换。消息代理服务器负责将业务的办理情况提交到监察平台，实现行政许可业务的监管。

（3）外网区域：为郊区县公证处用户和公众用户提供服务的区域，在线服务系统分别部署在应用服务器和数据库服务器上。SSL VPN为没有接入政务网络的下级单位以及公证处等提供安全的VPN服务，数据交换服务器用于通过互联网与已建系统公证处进行数据交换。

### 系统接口

本系统的接口包括内部接口和外边接口，内部接口主要是公证系统内部各个子系统业务模块间的调用和公证处综合子系统对相关设备驱动的调用；外部接口主要是公证系统与其他委办局和公证机构的数据共享服务。系统接口涉及到市局内网、政务专网和互联网三个网络，相关系统间的接口关系如所示。



**图 21 系统接口示意图**

（1）与已建系统公证处的接口：指的是本系统与已建系统公证处间的接口调用、数据共享和数据同步，比如：黑名单信息、不良记录、数据监管同步等。该方式通过消息中间件进行数据交换，保障数据的及时性和一致性。

（2）在线服务系统接口：为了满足公众实时查询办证情况和了解公证信息，需要把其相关情况同步到互联网，方便大众进行网上业务处理；

（3）办证系统与监管系统的接口：办证系统的信息，比如：办证收费、报表等信息需要实时同步到监管系统，监管系统也需要把考核、通告等信息发布到各个公证处，方便监管和交流；

（4）办证系统与外围设备的接口：办证系统涉及到一些设备，比如：二代身份证识别仪、扫描仪、条形码等设备，系统需要访问这些接口；同时，需要从公安局人口处获得当事人的照片和身份证信息，从建委获得当事人的房屋产权信息，从民政局获得当事人的婚姻登记信息，从银行获得当事人的资金信息；另外，监管系统通过电子地图查询各个公证处的地理位置、机构和人员基本信息，需要调用资源中心的电子地图接口，该方式通过Web Services技术进行实现。

### 框架设计

MVC（Modle-View-Controller）模式是Xerox PARC在20世纪80年代为编程语言Smalltalk-80所设计的一种软件设计模式，至今已被广泛使用，并被推荐为Sun公司J2EE平台的设计模式，其为开发交互式应用系统提供了一个优秀的设计模式，主要由三部分组成：模型、视图和控制器。Struts是Apache组织的一个开放源码项目，是一个比较好的MVC框架，提供了对开发MVC系统的底层支持，具有组件的模块化，灵活性和重用性的优点，同时简化了基于MVC的web应用程序的开发，它采用的主要技术是Servlet、JSP和Custom tag library等。

为了增加系统的复用度和提高开发效率，在Struts框架的基础上，建立了系统的FrameWork，系统FrameWork主要有视图层（表示层）、控制层、业务逻辑层、持久化层、数据库层和系统框架层六个层次，如所示。



**图 22 系统框架层次图**

（1）系统框架的MVC部分，主要应用Struts技术，同时开发了系统需要的自定义标签和样式表的控制功能，该技术在系统实现章节中结合公证机构的设立进行了介绍。

Struts框架中的Controller功能由ActionServlet和ActionMapping对象构成，核心是一个Servlet类型的对象ActionServlet，它用来接受客户端的请求。ActionServlet包括一组基于配置的ActionMapping对象，每个ActionMapping对象实现了一个请求到一个具体的Model部分中Action处理器对象之间的映射。

Model部分由Action和ActionForm对象构成，所有的Action处理器对象都是开发者从Struts的Action类派生的子类，Action处理器对象封装了具体的处理逻辑，调用业务逻辑模块，并且把响应提交到合适的View组件以产生响应。这里，对表单数据的验证是通过Struts提供的ActionForm组件对象来实现的，通过应用该技术提高了页面代码的规整性和可维护性。

View部分是通过JSP技术实现的。Struts提供了自定义的标签库，这些自定义标签可以很好地和系统的Model交互，通过使用这些自定义标记创建的JSP表单，可以实现和Model部分中的ActionForm的映射，完成对用户数据的封装。系统就是应用这些技术，实现了列表的排序、数据变更后保存提醒等功能。另外，该层的样式表技术也得到了很好的应用，提高了系统的美化效果。

（2）持久化层包含系统中的业务实体对象，与逻辑层一起对应分析模型中的业务模型。完成从业务对象到关系数据库的透明存储，使用对象关系映射工具hibernate来实现。数据访问对象实现了对hibernate和jdbc的封装，并且协调hibernate和jdbc两种数据访问接口，整个系统可以根据实际情况灵活选用不同方式。数据访问对象提供对象持久化功能和对象查询功能。对于一些很特殊的情况，需要直接访问hibernate或jdbc操作数据库。DAO与数据源管理的结合，使得通过直接访问hibernate或jdbc与使用DAO完全兼容，数据对象会自动同步，因此DAO具有很强的扩展性。

（3）数据库层采用关系型数据库，使用Oracle数据库管理系统来存储和管理数据。分为四个数据库，分别是基础库、业务库、共享库和在线库。

（4）系统框架层提供操作的封装，对系统的各逻辑分层提供支持。比如事务处理、异常处理和日志操作等。

由于系统采用了MDA建模技术，系统设计人员只要根据需求，建立系统的业务模型，就可以生成数据库和业务逻辑代码；而Web应用部分的开发，主要是针对用户需求，利用Struts的相关技术和框架实现视图层，对映射关系进行配置，在Action中调用建模人员生成的业务逻辑接口就可以了。

### 本章小结

本章详细分析了系统的体系架构、功能结构、网络拓扑、接口和框架设计方案，提出了系统的总体建设思路。本章的总体架构设计是系统的总体建设方案，也是详细设计和实现的基础和依据。

## 系统详细设计

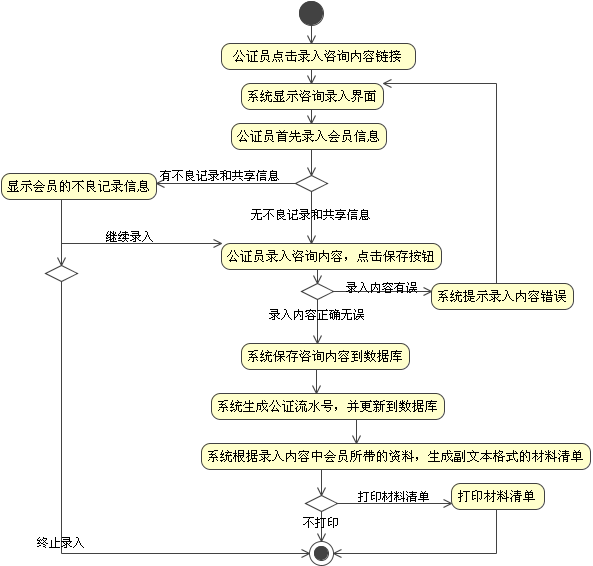
详细设计与实现主要依据系统总体架构设计，按照面向对象的理论方法分析了系统的活动、对象和时序关系等，并利用MDA工具构建了数据库，实现了公证处综合子系统、公证监管业务子系统、在线服务子系统的功能。过程中，通过对主要模块进行分析，阐明了设计和实现方法符合软件工程理论，对满足用户需求，发挥系统性能具有很重要的意义。

### 主要模块设计

公证咨询和机构设立是机构综合业务和监管业务系统的重点模块，下文将对其详细设计和实现情况进行重点描述，从静态和动态方面分析了功能的活动图、类图和时序图，为系统的实现奠定了基础。同时对办证模块的对象进行了梳理和抽象，建立了对象模型，详细介绍了公证咨询和受理模块的对象间的关系，对工具生成的办证模块的E-R图进行了展现，最后从系统的环境配置、系统实现技术、和系统界面等方面介绍了系统的实现情况。

#### 公证咨询模块

活动图（activity diagram，动态图）阐明了业务用例实现的工作流程，用于研究实现业务目标时所要执行的各项任务或活动的顺序安排，说明了业务为向所服务的业务主角提供其所需的价值而必须完成的工作，描述了业务用例中，用户可能会进行的相关活动。在2.3.1公证咨询中对业务用例进行了分析，活动图需要基于业务用例对每一个业务活动进行分析，其活动如所示。



**图 23 办证咨询模块活动图**

（1）公证员点击“咨询登记”菜单，系统显示咨询登记录入页面。

（2）公证员录入咨询人员信息。

● 公证员录入会员的唯一Id，系统自动将会员的身份证号、姓名显示出来。

● 公证员录入会员的身份证号/企业组织机构代码，系统自动将会员Id，姓名显示出来。

● 公证员通过第二代身份证扫描仪扫描身份证信息，系统自动填写身份证，并自动显示会员Id、姓名。

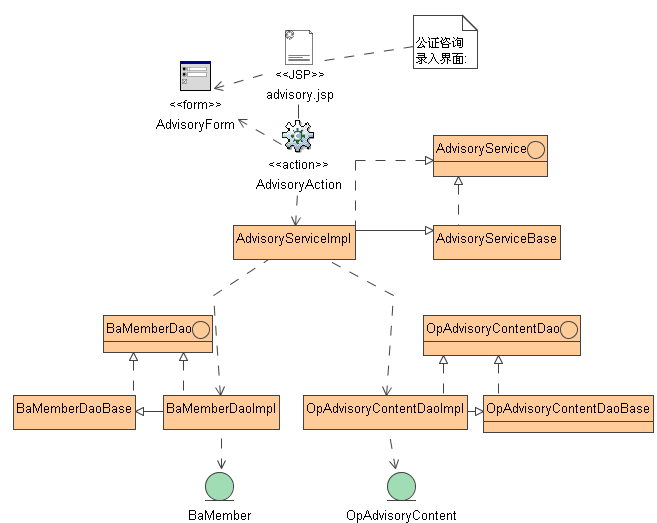
● 公证员点击“选择会员”按钮，根据查询条件，选中会员，返回记录咨询内容页面，会员ID、姓名/单位名称、证件类型、证件号码自动显示，然后点击“增加按钮”，保存后，返回记录咨询内容页面。

（3）系统根据会员的身份证号码搜索全市范围内是否具有不良记录、遗嘱公证、抵押公证。如果有不良记录、遗嘱公证、抵押公证，则系统在页面中间显示这些内容，可以点击关闭按钮使此信息消失。

（4）公证员点击保存按钮，系统将咨询信息保存，并生成公证流水号。

（5）系统显示保存后的信息的详细页面，页面上包括咨询信息的详细信息/公证流水号、“选择模板”下拉框、“生成材料清单”。

面向对象设计的基础就是使用类，类是用来代表现事物或者功能的构造块。它是由若干类关联组合在一起的，反映系统或者子系统组成结构的静态图，它由类和关联两个元素组成。办证咨询模块的类图关系如所示。

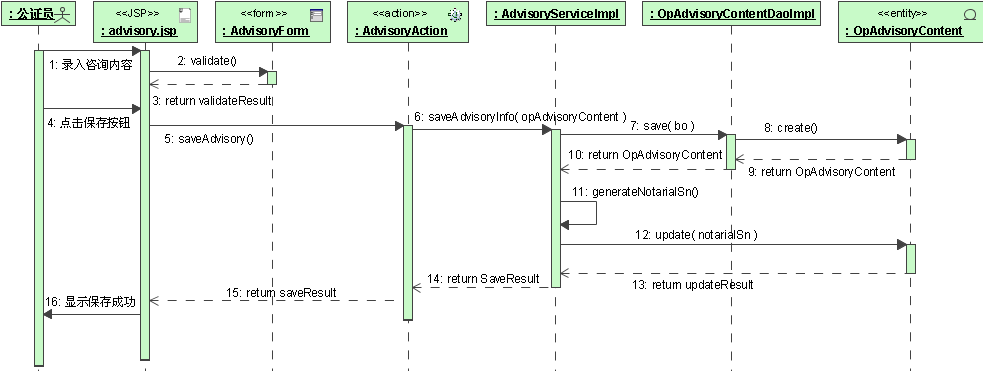


**图 24 办证咨询模块类图**

办证咨询模块主要有AdvisoryForm、AdvisoryAction、AdvisoryServiceImpl、BaMemberDao、BaMemberDaoBase、BaMemberDaoImpl、OpAdvisoryContentDaoImpl、AdvisoryService、AdvisoryServiceBase、OpAdvisoryContentDao和OpAdvisoryContent

-DaoBase组成，表明了该模块相关类之间的关联关系。其中AdvisoryForm 负责表现层与业务逻辑层之间的数据传输；AdvisoryAction负责页面流转控制，继承自Struts的DispatchAction类，AdvisoryAction调用业务逻辑类AdvisoryServiceImpl中的方法；AdvisoryServiceImpl是AdvisoryServiceBase的子类，而AdvisoryServiceBase类的接口是AdvisoryService；业务逻辑类AdvisoryServiceImpl调用DAO类OpAdvisoryContentDaoImpl，而OpAdvisoryContentDaoImpl实现对实体类的操作；BaMember和OpAdvidoryContent是咨询模块相关的实体类，BaMemberDaoImpl依赖BaMember，OpAdvisoryContentDaoImpl依赖OpAdvidoryContent。

UML 交互图以图形的形式表示出方法调用过程，它有两种形式：序列图（Sequence Diagram）和协作图（Collaboration Diagram）。序列图用于描述对象组如何随着时间在某些行为方面进行协作。序列图捕获单一用例的行为，同时显示在特定用例的时间框架中的对象以及这些对象之间传递的消息，但并不显示对象之间的关系。它以图的形式详细描述了业务角色和业务主角之间的交互，以及如何在执行业务用例时访问业务实体。序列图简要描述了参与业务角色的工作，从激活的角度描述了如何操作业务实体，以及它们如何通过发送消息实现通信。

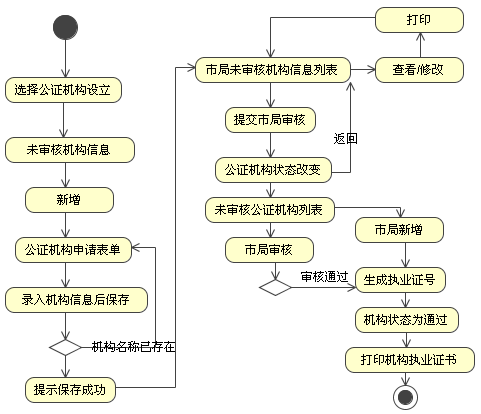


**图 25 办证咨询模块时序图**

有了静态结构，还需要确定系统的动态结构，这样，才能明确系统间的类是如何交互的，从而更好地帮助编码人员进行编码，这里借助时序图来展现，办证咨询模块各个类及方法间的时序关系如所示。该模块描述的是公证员在办理咨询业务时系统调用的相关文件、类中的方法及其返回结果之间的时序关系。主要过程是公证员先录入咨询内容，表单验证后反馈结果，保存后调用Action类， Action 调用业务逻辑类中的方法saveAdvisoryInfo，然后通过实体类OpAdvisoryContent保存到数据库中，最后反馈数据的保存情况。

#### 公证机构设立

区县司法局负责组建本区县管辖范围内的公证机构，并将组建公证机构的相关材料报市局进行审批。市局审核通过后，准予设立，并给公证机构颁发公证机构的执业证。系统实现公证机构信息的录入、审核、打印公证书、补全公证信息的功能。机构设立模块的活动如所示。



**图 26 公证机构设立模块活动图**

通过的活动图可以看出公证机构设立模块的具体活动如下：

（1）区县负责公证机构设立人员选择“公证机构设立”操作，系统显示本区先已经录入且未经过市局审核的公证机构信息列表，区县负责人选择“新增”操作，系统显示“新增公证机构信息”界面，界面内容见数据项部分。

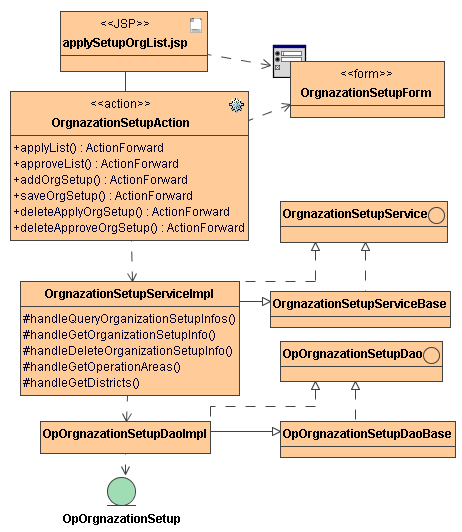
（2）区县负责公证机构设立人员录入界面中的数据项，选择“保存”操作。系统判断公证机构名称是否在系统中存在，如果已经存在，则系统提示“公证机构名称已经存在”；如果不存在，系统继续进行下一步。系统保存公证机构信息后，提示保存信息成功。

（3）区县负责公证机构设立人员选择“公证机构设立”操作，系统显示本区县已经录入且未经过市局审核的公证机构设立申请信息列表，用户任意勾选一条信息，点击“查看/修改”按钮，进入公证机构设立申请信息页面，用户点击“打印”按钮，系统弹出公证机构设立申请打印界面，点击打印图标，打印成功。

（4）区县负责公证机构设立人员选择“公证机构设立”操作，系统显示本区先已经录入且未经过市局审核的公证机构信息列表，区县负责人选择其中的一条公证机构设立信息，选择“上报”操作，系统将公证机构设立信息状态设置为上报，系统返回公证机构设立信息列表，列表中的所有信息均为未上报状态。

（5）市局负责公证机构设立人员选择“审核公证机构设立”操作，系统显示所有去区县司法局上报的公证机构设立信息列表，市局负责人选择一条公证机构设立信息，选择审核操作，系统显示公证机构设立信息的审核界面，最上方是审核意见并附有审核通过和审核不通过操作按钮，下方显示公证机构设立信息的详细内容，市局负责人录入审核意见，并选择审核操作。

上文对公证机构设立的工作活动流程进行了描述，这些流程是实现机构设立业务所需要的活动过程。是机构设立模块详细的对象类间的关系。



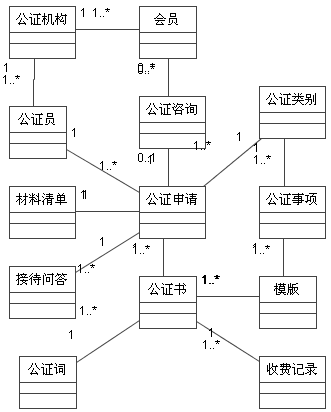
**图 27 公证机构设立模块详细类图**

该模块中有OrgnazationSetupAction和OrgnazationSetupServiceImpl类，OrgnazationSetupAction类中的方法有applyList()、approveList()等，该类的相关类是OrgnazationSetupForm类，它调用业务逻辑类OrgnazationSetupServiceImpl提供的方法，该模块相关的实体对象是OpOrgnazationSetup。

#### 数据库设计

在主要模块设计中通过UML图对功能的活动顺序和类间的关系进行了设计和明确，确定了类的相关属性和方法。在数据库设计方面，由于采用了MDA建模技术，所以在数据库设计时，首先需要抽取对象，明确对象间的关系，以及对象的属性和方法，这个抽象过程，更多地从平台无关性方面进行考虑，也就是不需要关心具体的技术实现。对象和关系确定后，也就完成了MDA的建模过程，接着利用AndroMDA生成数据库和部分程序。下文将按MDA的设计思想对办证模块进行设计。

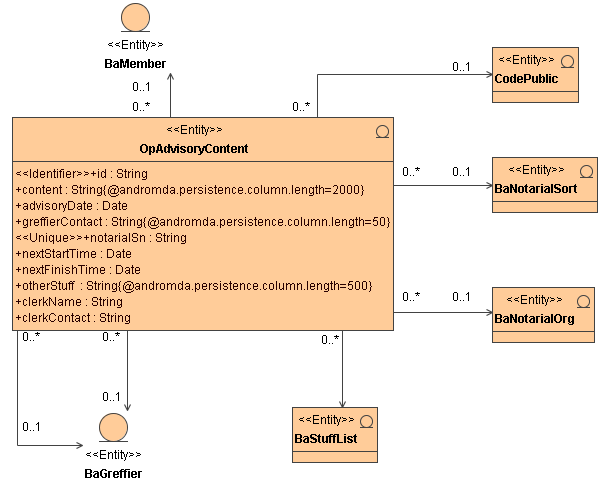
通过对办证模块的分析和抽象，建立了办证模块的对象模型，该模型有二十多个对象，办证模块的简单对象模型如所示。



**图 28 办证模块简单对象模型图**

办证系统中主要涉及到会员、公证机构、公证员、公证咨询记录、类别、材料清单、受理内容、公证事项、接待问答内容、公证书、模版、公证词和收费记录等对象。机构的公证员办理会员的公证申请时，首先进行公证咨询，然后才进行公证受理，对于受理需要明确公证类别和事项，同时告知相关材料清单，然后就是进行公证书的制作过程了，公证书、公证事项和公证词都涉及到模版，最后进行缴费。明确了对象和关联关系，接下来就需要结合业务对对象的属性和方法进行细化。实践证明，良好的对象模型，提高了系统的健壮性和开发效率，增强了系统的维护性和可扩展性。

是对公证咨询和公证申请模块的对象模型进行了细化，增加了对象的属性和方法，确定了属性和方法的访问权限、数据类型、返回类型、主外键和参数等。



**图 29 公证咨询对象模型图**

公证咨询模块的对象模型间的关系如下：

（1）相关对象：BaMember（会员基本信息）、CodePublic（公用码表）、OpAdvisory-

Content（公证咨询）、BaNotarialSort（公证类别）、Greffier（公证员）。

（2）对象间的关系：

BaMember对象与OpAdvisoryContent为一对多关系，即一个会员可以进行多项公证咨询。

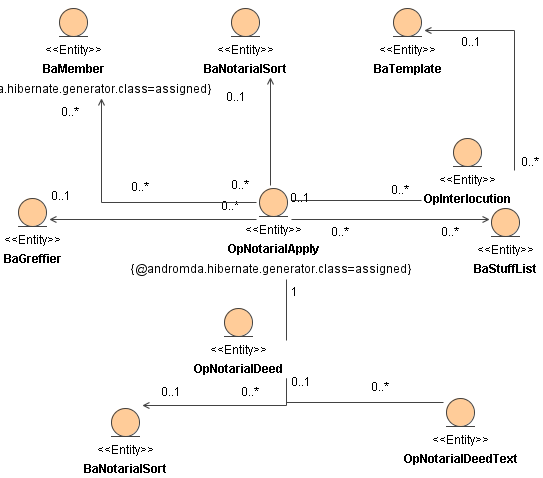
Greffier对象与OpAdvisoryContent为一对多关系，即一个公证员可以受理多项公证咨询。

BaNotarialSort对象与OpAdvisoryContent为一对多关系，即公证咨询中的咨询大类和咨询小类唯一。

CodePublic与OpAdvisoryContent为一对多关系，即公证咨询中的咨询用途和咨询方式唯一。

BaStuffList对象与OpAdvisoryContent为多对多关系。

公证申请模块的业务比较复杂，涉及的对象比较多，如所示。



**图 30 公证申请对象模型图**

公证申请模块的对象模型间的关系如下：

（1）相关对象：OpNotarialApply(公证申请对象)、BaNotarialSort（公证类别）、BaNotarialOrg（公证机构）。

（2）对象间的关系：BaNotarialOrg与OpNotarialApply为一对多关系；BaNotarialSort与OpNotarialApply为一对多关系。

通过对系统中对象模型的设计，抽象出了系统中已有的对象，明确了对象间的关联关系。这些对象和关系通过建模工具映射到了实体，实现了对象设计与数据库设计在面向对象体系上的一致。最后利用Maven，借助AndroMDA生成了数据库。这种设计方式与传统开发方式存在很大不同，传统开发方式需要基于需求数据项，对数据库直接的关联关系进行设计，然后建立数据库；而MDA建模技术主要针对数据对象，构建对象间的关系，通过工具，比如AndroMDA生成数据库和程序代码。

这里，只把办证业务核心的数据库E-R图进行展现，如所示。



**图 31 办证系统数据库E-R图**

数据库中的表结构主要包括字段名称、字段描述、数据类型、长度、是否为空、主键和外键等。是公证咨询的数据库表结构。

**表 1 公证咨询数据库表结构**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名称 | 字段描述 | 字段类型 | 长度 | 允许空 | 主/外键 |
| 1 | ID |  | VARCHAR2 | 255 |  | 主键 |
| 2 | CONTENT | 咨询描述 | VARCHAR2 | 2000 | √ |  |
| 3 | ADVISORY\_DATE | 咨询日期 | TIMESTAMP(3) | 11 | √ |  |
| 4 | GREFFIER\_CONTACT | 公证员联系方式 | VARCHAR2 | 50 | √ |  |
| 5 | NOTARIAL\_SN | 公证流水号 | VARCHAR2 | 255 | √ |  |

**表 2 公证咨询数据库表结构（续）**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名称 | 字段描述 | 字段类型 | 长度 | 允许空 | 主/外键 |
| 6 | NEXT\_START\_TIME | 下次来办理的起始时间 | TIMESTAMP(3) | 11 | √ |  |
| 7 | NEXT\_FINISH\_TIME | 下次来办理的结束时间 | TIMESTAMP(3) | 11 | √ |  |
| 8 | OTHER\_STUFF | 其他材料 | VARCHAR2 | 500 | √ |  |
| 9 | CLERK\_NAME | 接待人姓名 | VARCHAR2 | 255 | √ |  |
| 10 | CLERK\_CONTACT | 接待人联系方式 | VARCHAR2 | 255 | √ |  |
| 11 | MEMBER\_FK | 会员 | VARCHAR2 | 50 | √ | 外键 |
| 12 | ADVISORY\_ITEM\_FK | 咨询类 | VARCHAR2 | 255 | √ | 外键 |
| 13 | ADVISORY\_USE\_FK | 公证用途ID | VARCHAR2 | 255 | √ | 外键 |
| 14 | ADVISORY\_MODE\_FK | 咨询方式ID | VARCHAR2 | 255 | √ | 外键 |
| 15 | GREFFIER\_FK | 公证员ID | VARCHAR2 | 255 | √ | 外键 |
| 16 | ORG\_FK | 公证机构ID | VARCHAR2 | 255 | √ | 外键 |

，交互性强，达到了预