

基于软件体系结构的电子政务系统的研究

Study on Information System of Government based on Software Architecture Technology

常炳国 (湖南大学软件学院 长沙 410082)

摘要:数据、信息和知识集成是信息系统建设和应用的重要研究领域。本文从数据级、应用级和门户级三个层次阐述应用集成技术的重要作用 and 具体方法,并有效应用于实际系统。实践表明,采用软件体系结构不仅很好地满足使用者的现实需求,而且为信息系统的完善、扩展和升级开辟了很好的实践途径。

关键词:软件体系结构 应用集成 电子政务

计算机技术、网络通讯技术和现代控制技术已经渗透到政治、经济以及社会生活的各个领域,互联网应用已经从单一的信息浏览发展到复杂的分布式应用系统。政府各部分已经根据本部门的业务需求出发,建设和应用了多种形式的办公业务信息化系统,不同程度地提高了本部门的办事效率,成为电子政务建设和应用的重要。但是,随着信息化技术的迅速发展和人们日益增长的业务需求,需要借助于计算机完成那些具有充分想象力的高效率的服务理念、管理思想和人性化考核办法。这样就会引出一个问题:是全盘否定已经发挥作用,并将继续发挥作用的原有系统呢,还是在充分利用原有系统已积累的大量数据的、信息的和知识的重要资源,在此基础上实现更加完美的系统功能呢?显然后者是广泛接受的。

1 政府业务模型

政府职能可归纳为行政管理与服务、行政监督和政府资源管理与调控三大部分,由此产生了三个方面的政务业务活动:(1)上下级部门间的业务活动:依据国家的法律、法规,政府上级根据行政指导或命令对下级部门的管理与调控,实现社会、经济、文化资源的合理调配,同时积极响应下级部门的需求,接受监督;(2)社会实体管理服务活动:政府部门加强对社会实体的管理,把经济调节、社会管理、公共服务和公众信息作为政府的公共产品为社会实体提供服务。同时积极响应社会实体的服务请求并提供服务;(3)政府资源管理活动:资源是支撑政务

业务活动的前提,科学管理与合理开发利用人力、财力、物力资源是政务活动的重要组成部分。基于上述分析,设计政务业务模型如图 1 所示。

决策支持:在国家政策、法规指导下,对服务管理、办公管理、协同管理、信息资源管理等政府业务活动进行宏观调控与约束,构成政务业务活动的控制层;**服务管理:**在社会资源环境、办公管理、专项协同和信息资源管理支撑下,实现政府面向社会实体和公众的管理服务职能,提供各项服务。服务管理主要包括信息发布、信息采集、参政议政和为民办事等。**办公管理:**政府满足服务需求的具体实现手段。一方面,办公管理实现了政府上下级之间的政务管理活动,另一方面实现了部门对外的服务请求提供服务;**协同管理:**政府履行各项职能过程中需要多个专项业务系统的支持,专项协同实现了政务管理与各个专项业务之间协同工作,构成协同工作的环境;**资源管理:**实现了政府信息的共享与开发利用,并适应信息类型动态变化的要求,主要包括:1)信息资源分类编码:按照统一分类原则对信息进行分类,为信息的分类管理奠定基础;2)信息资源动态模型:按照统一信息资源模型定义信息资源(人力、财力、物力)的动态属性,形成各类信息资源动态模型;3)信息资源数据管理:对各类信息资源数据进行采集、整理加工、检索、维护等;4)信息资源模型转换:对已开发的信息资源(专业信息资源库、政务信息资源)进行格式转换。

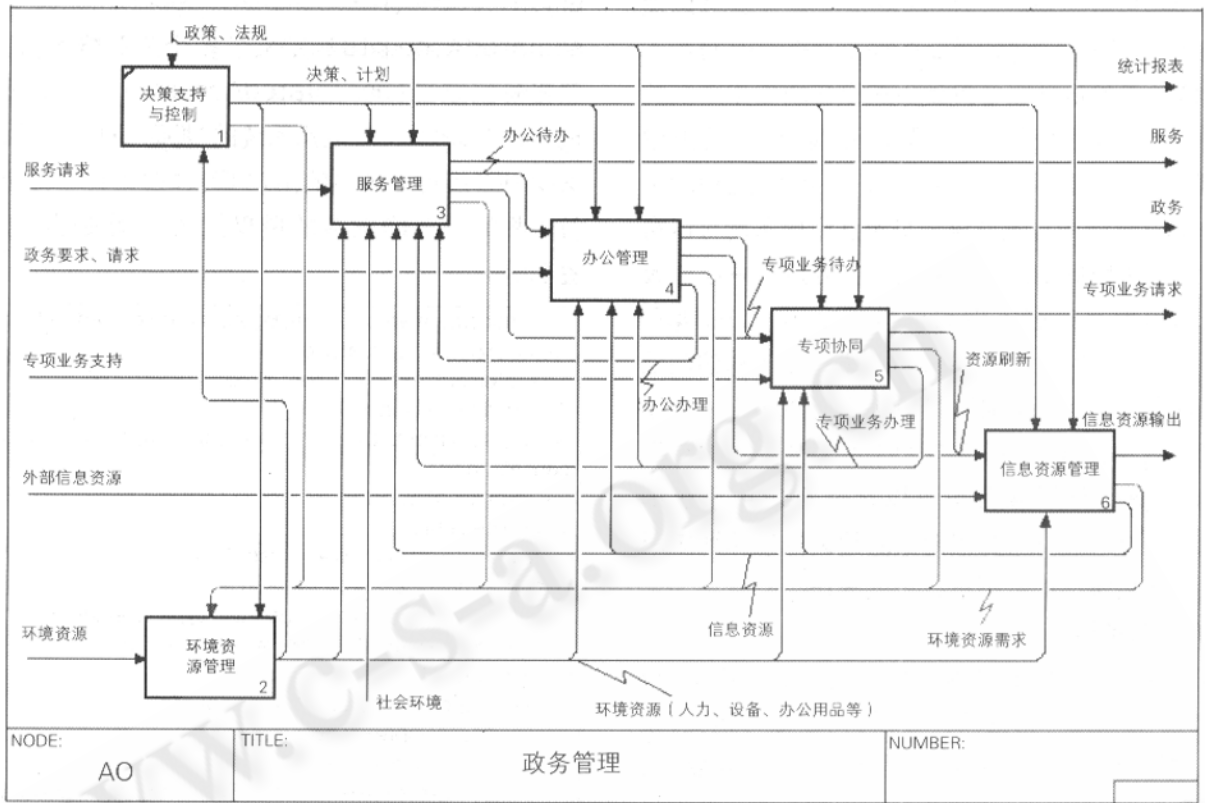


图 1 业务模型

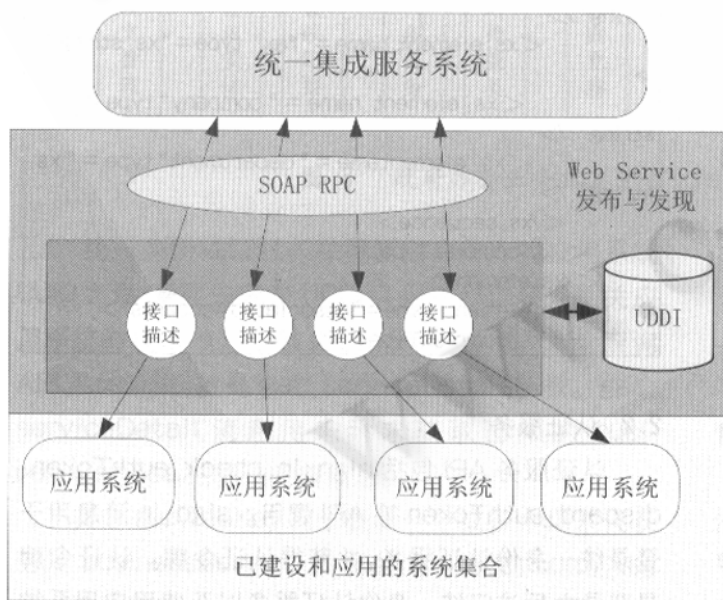


图 2 应用集成实现过程

2 应用集成

本文基于 Web Service 技术阐述应用集成过

程,其核心思想是:把原有应用封装为标准服务接口部署在 Web Service 服务器上,客户端应用程序通过 SOAP 调用已封装的服务接口,实现对原有应用的使用,其调用过程为:(1)客户端通过 HTTP 按照 SOAP 格式将发送请求(2)UDDI(Universal Description Discovery Integration)分析请求,发现服务接口的目标地址;(3)将目标服务和客户请求绑定,并对服务进行访问,接受访问反馈结果;(4)客户端接受的结果并展现。应用集成过程如图 2 所示。

基于统一身份识别服务的应用环境中,支持统一身份识别服务的应用系统以 UDDI 数据实体 businessService 的形式注册于 UDDI 中,并分配一个 UUID,用于标识相应的应用系统。businessService 是 UDDI 中的一个关键的数据结构,将一系列有关商业流程或分类目录的 Web Service 的描述组合到一起。businessService 和下面要提到的 bindingTemplate 一起构成了应用服务。内部应用环境使用 Private UDDI Registry(私有

UDDI 注册中心); Internet 外部应用环境可以考虑使用 Public UDDI Registry(公共 UDDI 注册中心)。

当统一身份认证服务得到用户提交的应用系统的 UUID 之后,它应当使用这个 UUID 查询 UDDI 注册中心,以获得这个应用系统的完整服务描述(businessService 结构),随后统一身份认证服务就可以得到这个应用系统的访问入口并访问。当应用系统修改了访问入口或是某些关键约束描述后,可以在 UDDI 中更新信息,以便再次关联查询时,获取更新的访问信息。这种方式可以无需修改代码实现动态绑定。为了提高服务的响应效率,减少与 UDDI 注册中心的交互次数,统一身份认证服务也可以选择将应用系统的 UDDI 描述缓存在本地,当用户访问相关的应用系统时可以直接使用缓存的数据进行后继操作。

2.1 Web 服务接口

统一身份认证服务的对外接口由两部分组成:1) 用户 Profile 维护:包括用户注册,关联帐号定义,帐号信息维护等。2) 认证服务:包括用户登录/注销,认证令牌认证,访问转发,访问代理等。用户 Profile 维护的 XML Schema 定义代码片段如下:

XML Schema 使用 save_user API 和 delete_user API 维护用户帐号。save_user 用于新建帐号和更新帐号。当统一身份认证服务接受 save_user API 调用时,若发现服务的用户库中并没有相应的 userID,就新建帐号;若发现服务的用户库中有相应的 userID,那么就是更新帐号。而 delete_user 则用于删除帐号,当然前提是服务的用户库中有相应的 userID,否则就是非法调用。

delete_user API 调用中,顶级元素是 delete_user,delete_user 有两个子元素 userID 和 credential,userID 是用户的标识,其类型是 email 地址。而 credential 则是对应的授权信息,可能是用户密码,数字签名等形式。

save_user API 顶级元素是 save_user,而 save_user 有一个子元素 user,user 元素完整定义了一个用户帐号的详细信息。除了 userID 和 credential 子元素外,user 元素还包括两个复合元素 personalInfo 和 associatedAccounts。personalInfo 元素描述了用户的个人信息,包括: name(姓名)、alternativeEmail(可替换的 email,userID 是其使用的主要

email)、phone、fax、company、department。而 associatedAccounts 则定义了多个关联帐号,每个关联帐号的定义(associatedAccount)由三个元素组成:applicationID(应用系统的 UDDI UUID 标识)、accountID(应用系统中的用户 ID)、credential(应用系统中相关用户 ID 对应的授权信息,可能是密码、数字签名等)。

```
<xs:schema elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:element name="user" type="userType">
    <xs:annotation>
      <xs:documentation> USASUserDefinition </xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:complexType name="userType">
    <xs:sequence>
      <xs:element name="userID" type="emailType" />
      <xs:element name="credential" type="xs:base64Binary" />
      <xs:element name="personalInfo">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="name" type="xs:string" />
            <xs:element name="alternativeEmail" type="emailType" />
            <xs:element name="phone" type="xs:string" />
            <xs:element name="fax" type="xs:string" />
            <xs:element name="company" type="xs:string" />
            <xs:element name="department" type="xs:string" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="associatedAccounts">
        <xs:complexType>
```

2.2 认证服务

认证服务 API 包括 sign_in、check_authToken、discard_authToken 等 API 调用。sign_in 消息用于登录统一身份认证服务,并获得认证令牌。认证令牌是登录之后访问统一身份认证服务以及使用应用系统所必须的不可缺少的必要参数。当 sign_in 成功调用,统一身份认证服务需要返回认证令牌,以供后续授权访问。authToken 消息是 sign_in 消息的响应消息。它返回认证信息 authInfo,认证信息将用于后续的调用。authToken 消息包含单个 authInfo 元素,该

元素包含一个访问令牌,令牌将在所有后继 API 消息调用中被使用。作为一个对 sign_in 消息的同步响应,这个消息返回时始终使用 SSL 加密。当应用系统收到用户传来的认证令牌后,可以选择到统一身份认证服务验证该认证令牌的合法性,并依据结果判断是否接受用户的访问请求。这个验证认证令牌的 API 函数是 check_authToken。

当认证令牌合法,将返回 valid,如果认证令牌不合法或者已经失效,则返回 invalid。当用户完成了它所需的各种操作,那么它应当从统一身份认证服务中注销,以避免获取的认证令牌被非法滥用。此时它可以使用 discard_authToken 函数。discard_authToken API 调用用来通知统一身份认证服务可以丢弃某个认证令牌,也就是终止当前的会话。以后一切使用该认证令牌的调用全部会被拒绝。

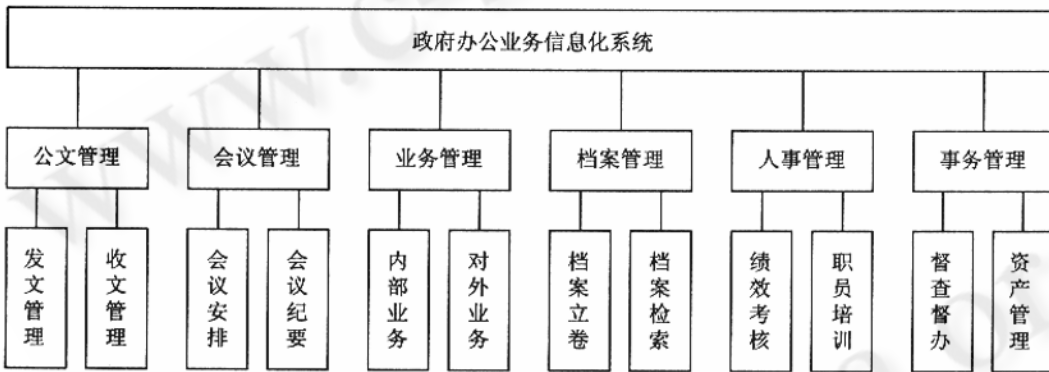


图 3 政府办公业务信息化系统功能结构

统一身份认证服务在得到用户传入的应用系统 UUID 之后,需要去查询 UDDI 注册中心以获得相关应用系统的技术信息以被之后的服务调用。这个查询 API 具体来说,就是 get_serviceDetail 函数。get_serviceDetail 消息用来请求获取已知的 businessService 结构的完整信息。

其中 serviceKey 用于表示一个或多个用来代表已知 businessService 数据特定实例的 uuid_key 值。该函数若成功调用,则返回一个 serviceDetail 消息,其成功匹配了一个或多个 serviceKey 值。如果传入的是多个 serviceKey 值,其结果将根据传入的值的次序依次返回。如果匹配到的记录数量过多,操作入口站点将对返回值执行截断操作。如果该情况发生,serviceDetail 将包含一个 truncated 属性,且

该属性值为 true。serviceDetail 消息返回一个或多个完整的 businessService 结构,作为对 get_serviceDetail 查询消息的响应。

3 实例应用

政府办公业务信息化系统是政府各部门业务信息和办公信息的收集与处理、交换与共享、综合利用与科学决策的具有战略意义的业务管理系统。该系统主要功能包括:公文管理、会议管理、业务管理、档案管理、人事管理、事务管理等,其功能结构如图 3 所示。

在上述分析基础上,设计了公文交换以及发文、收文流程管理。电子公文采用 XML(eXtensible Markup language)标准格式实现公文管理。公文交换的主要内容是:①建立电子公文交换的技术规范;②建立政府各部门公文上传下达的交换平台。

4 结束语

电子政务系统将在政务活动中发挥越来越重要的作用,开放式组件技术将大大加快开发效率,使政务办公自动化管理系统具有动态可扩展性、支持分布式和易维护性,有效地提高系统的灵活性。论文详细论述了

公文交换和发文、收文过程,从而为电子政务的发展与应用展开有意义的研究。

参考文献

- 1 王德禄,知识管理的 IT 实现 [M],北京:电子工业出版社,2003,4.
- 2 Dr Andreas Vogel. The Inprise Application Server Building Enterprise Application for the Net with EJB, CORBA, and, XML. 1999. <http://www.inprise.com>[J].
- 3 WfMC. Workflow Management Coalition Specification: Terminology & Glossary. Document Number WfMC - TC - 1011, Brussels, 1996 [J].