

基于软构件技术的 OA 系统的设计与实现

段润英¹, 薛锦云^{1,2}, 吕明琪¹

- (1. 江西师范大学 计算机信息工程学院, 江西 南昌 330027;
2. 中国科学院 软件研究所 计算机科学重点实验室, 北京 100080)

摘 要:实现办公自动化是提高办公质量和效率,促进政务公开的一个重要手段。但是由于企业或单位的办公实际环境和需求千差万别,使得当前很多办公自动化系统存在开发周期过长、重复投资、可适应性和可维护性差等问题,无法满足用户需求。详细说明了基于构件的 OA 系统的正交结构模型,并阐述了基于构件的 OA 系统的实现方法。基于构件的 OA 系统具有敏捷性,能够随着办公业务的变化随时调整,以满足不同政府、企业的不同需求。同时它还具有良好的可扩展性、可维护性和可重用性。

关键词:OA 系统;软构件;正交;Spring

中图分类号:TP317.1

文献标识码:A

文章编号:1673-629X(2007)08-0095-04

Design and Implementation of OA System Based on Software Component

DUAN Run-ying¹, XUE Jin-yun^{1,2}, LÜ Ming-qi¹

- (1. College of Computer Information Engineering, Jiangxi Normal University, Nanchang 330027, China;
2. Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing 100080, China)

Abstract: The implementation of an office automation system helps to improve quality of service and efficiency and to improve the progress of building an openness government. But OA system has witted a profound change in scale, structure, function and design fields. So that appears many problems, such as long development cycle, repeated investment, low maintainability etc. So it is impossible to build such software systems because of the diversity of application environment and demands. This paper introduces an orthogonal architecture model of the OA system based on component. The method of implementation of the office automation system application is discussed in this paper. The OA system based on component is easier to be changed to satisfy the government's requirement. It is extensible, maintainable and reusable.

Key words: OA system; software component; orthogonality; Spring

0 引言

随着信息技术的发展,办公自动化已成为政府、企业发展的必然趋势。办公自动化系统作为 Intranet 的一部分,正受到越来越多政府、企业的青睐。然而由于以下几种原因使得各政府、企业开发办公自动化系统时都得“一切从零开始”:①各政府、企业的办公业务不断变化,千差万别;②传统的办公自动化系统由于采用了刚性结构,系统从高层到低层的功能都固化在系统

程序中,当其中一个层次发生变化时系统就要重新开发,不利于系统功能的调整和扩展。传统办公自动化系统一经开发完成就具有相对稳定性,难以满足办公业务变化的需要。

由于各政府、企业办公业务千差万别和不断变化,对办公自动化系统的可扩展性、可维护性和可重用性提出了很高的要求。办公自动化系统的扩展、维护和重用可能表现为:增加新的办公业务功能;业务功能的局部改动;将第三方办公自动化系统的部分功能模块作为一个模块融入到本系统中来。

从软件设计的角度来看,近十几年来日益被重视的软件构件技术正能满足办公自动化系统的这种需求。软件构件是可复用的软件组成成分,可被用来构造其他软件。基于软件构件的程序设计的基本思想

收稿日期:2006-11-06

基金项目:国家自然科学基金资助项目(60273092)

作者简介:段润英(1982-),女,江西峡江人,硕士研究生,研究方向为软件工程;薛锦云,教授,博士生导师,研究方向为软件形式化与自动化、软件工程。

是:将软件中的功能部分分解为各个独立的构件,这些构件与外部仅通过预先定义的接口进行通信。在办公自动化系统中利用软件构件技术,可以实现系统的可重构性、扩展性、重用性和组装性。

1 基于构件的 OA 系统

1.1 构件的基本概念

在软件工程领域中,构件概念的范围是广泛的,没有统一的定义。笔者倾向于文献[1]中关于基于构件的软件工程的报告中对构件的定义:构件是一种功能上不透明的实现,能够被第三方组装,是由构件模型构造出来的。不透明功能实现的要求主要出于两个方面的原因:①对于构件市场的考虑,基于主导地位的商业构件模型必须通过隐蔽实现细节来保护;②构件的应用客户不需知道内部实现细节。可被第三方组装的要求是直观的:作为一个可独立发布的软件实体,应具有复用的价值,构件的使用不能仅仅依靠于提供者的工具或知识。最后,构件模型负责构件之间的交互方式以及全局的或体系结构上的限制。

1.2 系统模型

实际上,尽管各政府、企业办公业务千差万别,但具体的业务功能总是有限的,由一些相对稳定的管理单元构成,每个管理单元可以视为一系列管理与决策的活动。在办公自动化系统中,业务活动由一些基本稳定的功能单元组成,这些功能单元又由更低层次的功能单元构成,最终表现为对数据的操作。在系统设计中,根据办公自动化系统的层次性和功能性,采用了层次化多级正交软件体系结构,这种结构的基本思想是把应用系统的结构按功能的正交相关性,垂直分割为多个线索,线索又分割为几个层次,每个线索由多个具有不同层次功能和不同抽象水平的构件构成^[2]。线索是子系统的特例,它是由完成不同层次功能的构件构成,每一条线索完成系统中相对独立的一部分功能^[2]。对于大型复杂软件系统,每一条线索又可以按类似的方法划分为若干条子线索,其子线索还可以划分为更低一级的子线索,形成多级正交结构。如果线索间是相互独立的,即不同线索中的构件之间没有相互调用,那么这个结构就是完全正交的^[3]。由于完全正交结构不能很好地适用于本系统,故放宽了对结构正交性的要求,允许在线索间有适当的相互调用,因为各功能或多或少有相互重叠的地方,因此会发生共享某些构件的情况。根据层次

化多级正交软件体系结构的思想,从横向和纵向对系统进行正交划分,把不同层次不同功能的基本单元作为基本构件(原子构件)单独开发,通过对低层次不同功能的构件单元的组织实现系统功能的可重构、可扩展和组装。

图 1 是基于构件的办公自动化系统 OA 的结构示意。在该办公自动化系统中从横向(层次)将系统划分成四个层次:表示层、控制层、业务逻辑层、数据访问层。从纵向(线索)将整个系统划分为个人事务管理、工作流程管理、组织人事管理、公共信息管理、行政办公和系统管理五个子系统(主线索)。各个子系统再细分成粒度更小的功能模块(子线索),在功能模块划分时遵循以下原则以保证各功能模块的独立性:每一功能模块的实现与其他功能模块的实现无关或关联较少——正交划分原则。各个功能模块可看成一个复合构件,该复合构件可能由复合构件或各层次的不同功能的原子构件组装而成。

通过正交划分得到不同层次不同功能的若干原子构件。每个构件有其特定的接口和服务,且每个构件的接口都定义了其对外提供的功能和所需的功能。从层次上可分为四种类型的原子构件:Web 构件,控制构件,业务逻辑构件,数据访问构件。下面将介绍这四种原子构件。

Web 构件:主要用于显示数据,实现用户界面的功能。Web 构件通过控制构件(控制连接件)实现与低层次的业务逻辑构件的连接组装。

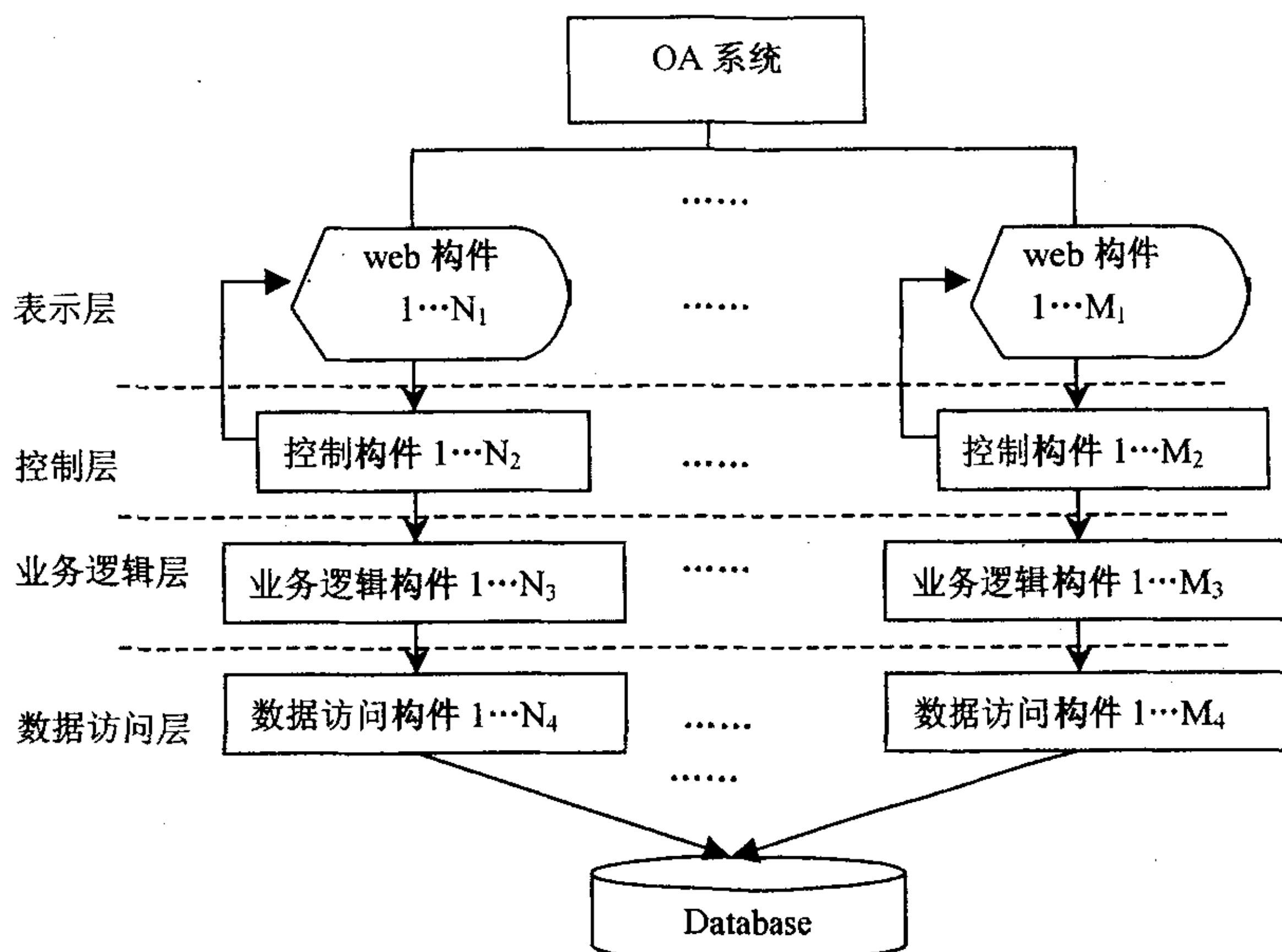


图 1 基于构件的办公自动化系统正交结构模型

控制构件:也叫控制连接件,定义了控制逻辑。控制构件根据需要调用业务逻辑构件完成各种处理方法,并将结果返回,导向相应的 Web 构件对视图进行显示。

业务逻辑构件:在业务逻辑构件中,根据需求的定义制定好各种业务逻辑,并调用相关的数据访问构件完成数据库操作。

数据访问构件:实现对数据库操作的功能,给高层次的业务逻辑构件提供服务。

各不同层次不同功能的原子构件通过不同的连接方式组装成大粒度的复合构件,再通过复合构件集成最终完成整个OA系统的开发。

2 系统实现

在OA系统中为了实现基于构件的软件结构采用了基于J2EE平台规范的Spring开发框架结构。OA系统的客户端主要由Web构件组成;用JSP技术进行开发,在Spring框架提供Spring Web MVC包、Spring-DAO包、SpringCore包和支持依赖注入技术的基础上实现Web构件、控制构件、业务逻辑构件、数据访问构件和各构件的组装,最终完成整个系统的开发。

2.1 Spring 框架

Spring是J2EE平台规范的一个实现框架。它提供了Dependency Injection机制的组件配置、灵活的MVC Web应用开发、整合JDBC、ORM多种数据访问技术等功能。按照文献[4],Spring Framework可以称为一个基于构件的软件框架。它为基于构件的软件开发提供了强有力的支持。其中依赖注入技术起源于轻量级构件容器提供的控制反转机制(Inversion of Control),即由容器定位插件的具体实现。依赖注入技术用部署描述文件描述构件之间的依赖关系,在运行时由容器按部署描述文件动态地将被调用构件注入到调用构件之中^[5]。所以,在OA系统中利用依赖注入技术在部署描述文件中对构件进行描述,并利用它更好地实现各构件的组装。

2.2 基于构件的OA系统实现

Spring技术框架是基于软构件技术的OA系统实现的底层基础。在Spring框架下构造出来的构件具备文献[1]中对构件的基本要求。在该系统中定义的构件实现类都遵循JavaBean规范,即所有的构件实现类都包括无参数的构造函数和相应的Setter方法。在完成构件的定义之后,本系统实现的另一个关键问题是如何实现各构件的组装。

如上文所述,采用了Spring Framework中的依赖注入技术。在Spring框架下通过构件的配置文件来管理OA系统中定义的构件,配置文件的内容包括构件的实现类、属性设置、合作者(被调用构件)声明等。然后,Spring Framework根据配置文件中定义的构件间的依赖关系(即上述的合作者),调用相应的Setter方

法,设置好合作构件(被调用构件)。在运行时由容器按配置文件动态地将被调用构件注入到调用构件之中,以此实现构件的组装。

OA系统中将行政办公子系统(主线索)进一步划分为:会议管理、车辆管理、资产管理、办公用品管理四个功能模块(子线索)。其中会议管理提供会议的申请、审批、会议室安排、会议通知、材料分发、会议回执、会议纪要起草、审批、分发的全过程管理,能够进行会议室的登记、预约、冲突检测等功能。下面以会议申请子模块为例阐述基于构件的OA系统的实现。

可以把整个会议申请子模块看作一个复合构件,该复合构件满足1.2节中阐述的结构模型。该复合构件是由Web原子构件ConferenceAppPage、控制原子构件ConferenceAppController、业务逻辑原子构件ConferenceOp和UserOp、数据访问构件ConferenceDao和UserDao组装而成。其中每一个原子构件都要实现其对外提供的接口并利用IoC模式在配置描述文件中描述其所需的外部接口,以原子构件ConferenceDao为例,ConferenceDao接口如下:

```
public interface IConferenceDao {
    void createConferenceInfo(ConferenceInfo info);
    Object getConferenceInfoByNo(String conferenceNo);
    Boolean updateConferenceInfo(ConferenceInfo info);
    .....
}
```

接口实现如下:

```
public void ConferenceDao implements IconferenceDao{
    private DataSource datasource;
    public void setDatasource(DataSource ds){
        this.datasource = ds;
    }
    public void createConferenceInfo(ConferenceInfo info){
    .....
    }
    .....
}
```

配置文件描述如下:

```
.....
<bean id="conferenceDao" class="net.sstc.webapp.oa.meeting.
dao.ConferenceDao">
    <property name="datasource">
        <ref local="datasource"/>
    </property>
```

在原子构件都已经得到实现的条件下,同样可以利用依赖注入模式对原子构件进行组装,组装过程可以在配置描述文件中进行描述如下:

```
<bean id="urlMapping"
```



```

class = "org.springframework.web.servlet.handler.SimpleUrlHan-
dlerMapping">
    <property name = "mappings">
        <props>
            <prop key = "/ConferenceApp.spring"> conferenceAp-
pController</prop>
        </props>
    </property>
</bean>
<bean id = "conferenceAppController"
class = "cn.edu.ssct.oa.meetingmodule.controller.ConferenceAp-
pController">
    <property name = "conferenceOp"> <ref bean = "confer-
enceOp"/>
</property>
    <property name = "userOp"> <ref bean = "userOp"/> <prop-
erty>
</bean>
<bean id = "userOp" class = "cn.edu.ssct.oa.meetingmodule.busi-
ness.UserOp">
    <property name = "userDAO"> <ref bean = "userDAO"/> </
property>
</bean>
<bean id = "conferenceOp"
class = "cn.edu.ssct.oa.meetingmodule.business.conferenceOp"
>
    <property name = "conferenceDAO"> <ref bean = "confer-
enceDAO"/> </property>
</bean>
<bean id = "userDAO" class = "cn.edu.ssct.oa.meetingmodule.
dao.UserDAO">
</bean>
<bean id = "conferenceDAO"

```

(上接第 94 页)

ogo 软件平台,利用了“Cooperation 模型”,仿真模拟了不同利用方式下有限资源下对经济发展的支持作用,获取了社会财富积累随时间变化的曲线,得到经济学启示。

多主体模型以并行的方式模拟非线性因果的社会经济系统,使人们更好地理解经济现象,发现现象背后的机制,从而做出预测和辅助决策^[6]。多主体建模目前还未形成成熟的体系,因而也没有一套完整而形成的理论,但可以预言,随着多主体思想的普及及理论方法的完善,基于多主体建模会越来越多地应用于社会经济研究中。

```

class = "cn.edu.ssct.oa.meetingmodule.dao.conferenceDAO">
</bean>

```

这样会议申请子模块在 Spring 框架下以构件的方式得到了实现。

3 结束语

详细说明了基于构件的 OA 系统的系统结构模型,该模型采用正交划分方式将 OA 系统看作四个不同层次不同功能的四种原子构件的组装和集成。通过正交划分实现的结构具有易理解性、自适应性、可重用性和结构易调整性^[2]。并阐述了如何在 Spring 框架下以构件的方式实现 OA 系统。基于构件的 OA 系统具有敏捷性,能够随着办公业务的变化随时调整,以满足不同政府、企业的不同需求。同时它还具有良好的可扩展性、可维护性和可重用性。

参考文献:

- [1] Bachman F, Bass L, Buhman C, et al. Volume II: Technical Concepts of Component - based software Engineering[M]. 2nd Edition. CMU, Pittsburgh:[s. n.],2000.
- [2] 谭凯,林子禹,彭德纯,等. 多级正交软件体系结构及其应用[J]. 小型微型计算机系统,2000,21(2):138-141.
- [3] Rajlich V, Silva J H. Evolution and reuse of orthogonal architecture[J]. IEEE Transactions on Software Engineering,1996(2):153-157.
- [4] 刘瑜,张世琨,王立福,等. 基于构件的软件框架与角色扩展形态研究[J]. 软件学报,2003,14(8):1364-1370.
- [5] Fowler M. Inversion of Control: Containers and the Dependency Injection Pattern[EB/OL]. 2004-01-23. <http://martinfowler.com/articles/injection.html>.

参考文献:

- [1] 张江,李学伟. 人工社会——基于 Agent 的社会学仿真[J]. 系统工程,2005(1):23-26.
- [2] 唐建荣. 生态经济学[M]. 北京:化学工业出版社,2005.
- [3] 方美琪,张树人. 复杂系统建模与仿真[M]. 北京:中国人民大学出版社,2005.
- [4] Epstein J M, Axtell R. Growing artificial societies, social science from the bottom up[M]. Washington D C: Brookings Institution Press; Cambridge, Mass: MIT Press,1996.
- [5] Wilensky U. NetLogo Utilizer Manual 3.1[EB/OL]. 2005. <http://ccl.northwestern.edu/netlogo>.
- [6] 王飞跃,兰森 S. 从人工生命到人工社会——复杂社会系统研究的现状和展望[J]. 复杂系统与复杂性科学,2004(1):48-51.