

# J2EE Web 开发框架体系结构

杜小刚<sup>1</sup> 李舟军<sup>1,2</sup>

(国防科技大学计算机学院 长沙 410073)<sup>1</sup> (北京航空航天大学计算机学院 北京 100083)<sup>2</sup>

**摘 要** 开源框架(如 MVC 框架 Struts、OR Mapping 框架 Hibernate、Log 框架 Log4j 等)的出现极大地提高了 J2EE 应用程序开发的效率,但它们都只提供了应用程序某一层次的框架,不是一个完整的应用框架。应用框架是整个系统的可重用设计,是构建应用程序的模板,它本质上是一系列设计模式的抽象实现,并提供一些框架基础服务。在整合各种框架的基础上,我们在一个更高的层面上设计和实现了一个 J2EE Web 开发框架。该开发框架具有良好的软件体系结构,采用了多种架构设计模式(如多层结构、MVC 模式、IoC 模式等),保证了程序具有松耦合性和易扩展性,并提供了一些常用的可复用构件,实现了 Web 应用系统的基础功能。它可以帮助开发人员获得最大程度的框架复用,快速开发应用系统。

**关键词** 软件体系结构, 开发框架, 框架复用

## Architecture of J2EE Web Development Framework

DU Xiao Gang<sup>1</sup> LI Zhou Jun<sup>1,2</sup>

(School of Computer, National University of Defence Technology, Changsha 410073)<sup>1</sup>

(School of Computer Science & Engineering, Beihang University, Beijing 100083)<sup>2</sup>

**Abstract** Though the emerging open source frameworks such as the Struts MVC framework, the Hibernate OR Mapping framework, the Log4j log framework etc, have highly promoted the efficiency for developing J2EE application, they are only the frameworks for a certain application layer, not for all the application layers. The application framework is a reusable design of a system and a template of building an application. Essentially it is an abstract implementation of a set of design patterns and provides some infrastructure services. We design and implement a J2EE Web development framework in a higher level based on the integration of various kinds of frameworks. The development framework has a good software architecture and uses a variety of architecture design patterns, including n tier model, MVC pattern and IoC pattern. Accordingly the framework is loose coupled and extensible. Furthermore, it delivers some usually used reusable components and the basic system functions of a Web application. Thus, it can help software engineers to get the maximal framework reuse and quickly implement the application.

**Keywords** Software architecture, Development framework, Framework reuse

### 1 引言

回顾过去 20 年的软件技术发展,我们可以清晰地看出软件重用技术的发展脉络。从早期 C 语言类库函数的重用,到面向对象编程的迅速壮大,到可复用构件<sup>[1]</sup>的发展和成熟,再到最近几年应用框架的重用。软件重用的粒度越来越大,软件复用的效率越来越高。

Sun 公司推出的 Java2 Enterprise Edition(J2EE)平台<sup>[2]</sup>已经日趋成熟,J2EE 得到了广泛的应用。但是 J2EE 应用的开发比较复杂。在 J2EE 应用开发的最初阶段,每个人都需要写自己的框架,如 MVC 框架、持久化框架、日志框架等。这些框架的编写都要花费很多时间,却没有取得相应的效果,甚至导致了更加复杂的编程模型和更低性能。直到最近两三年,才出现了一些得到广泛认可的开放源码的框架,如 MVC 框架 Struts<sup>[3]</sup>、OR Mapping 框架 Hibernate<sup>[4]</sup>、Log 框架 Log4j<sup>[5]</sup>等,这些框架的出现极大地提高了应用程序开发的效率。但是,它们都只提供了应用程序某一层次的框架,不是一个完整的应用框架<sup>[6]</sup>。应用框架是整个系统的可重用设计,是构建应用程序的模板。本质上应用框架是一系列设计模式<sup>[7]</sup>的抽象实现,并提供一些框架基础服务。

在整合各种框架的基础上,我们在一个更高的层面上设计和实现了一个 J2EE Web 开发框架,该框架提供了一个一致的页面布局模型和统一的页面交互控制,还提供了一些常

用的可复用的基础构件和页面构件,并在此基础上形成了一套有效的应用开发模式,实现了 Web 应用系统的基础功能。开发人员可直接在该框架的基础上进行应用开发,集中精力实现应用的业务逻辑,而不必在技巧性要求较高的、复杂的基础框架上浪费时间和精力。本文所述的开发框架对于 J2EE Web 应用系统的开发具有如下的重要意义:

- 提高了软件生产率;
- 提高了软件产品质量;
- 缩短了开发周期;
- 降低了维护费用;
- 便于软件移植和实现互操作性;
- 支持快速原型开发;
- 具有良好的软件体系结构;
- 减少了对系统架构师的需求;
- 规范了应用的开发模式。

### 2 开发框架体系结构

开发框架的体系结构如图 1 所示。

开发框架采用典型的多层结构,分为客户端、表现层、业务逻辑层、数据访问层、数据层,各层次间相对独立和松散耦合,保证了使用开发框架的应用具有良好的结构。开发框架提供了一些常用的可复用构件,特别是在表现层,这些构件可以给开发带来很大的方便。开发框架中包含的页面框架使得

杜小刚 硕士生,主要研究方向为应用软件架构设计;李舟军 教授,博士生导师,研究方向为计算机科学理论、数据仓库与数据挖掘。

页面开发可以遵循统一的页面布局和使用统一的页面交互控制。另外, 开发框架集成了多种流行的开源框架, 如 Spring<sup>[8]</sup>, Struts, Hibernate, iBatis<sup>[9]</sup>, 提供了统一的异常处理和数据传输机制, 在此基础上还开发了常用的基础功能应用。这样的开发框架既利用了现有的优秀技术, 又在整合这些技术基础上形成了一套有效的开发模式, 使得新的 Web 应

用开发不需要从零开始, 而是可以直接在开发框架的基础上开发业务应用。开发框架同时具有较好的移植性, 支持多种应用服务器, 如 JBoss、Weblogic、Websphere; 支持多种数据库, 如 DB2、M S S Q L Server、Oracle、Informix; 支持多种操作系统, 如 Windows、Linux、Unix。因此开发框架可以在各种不同环境的项目中获得应用。

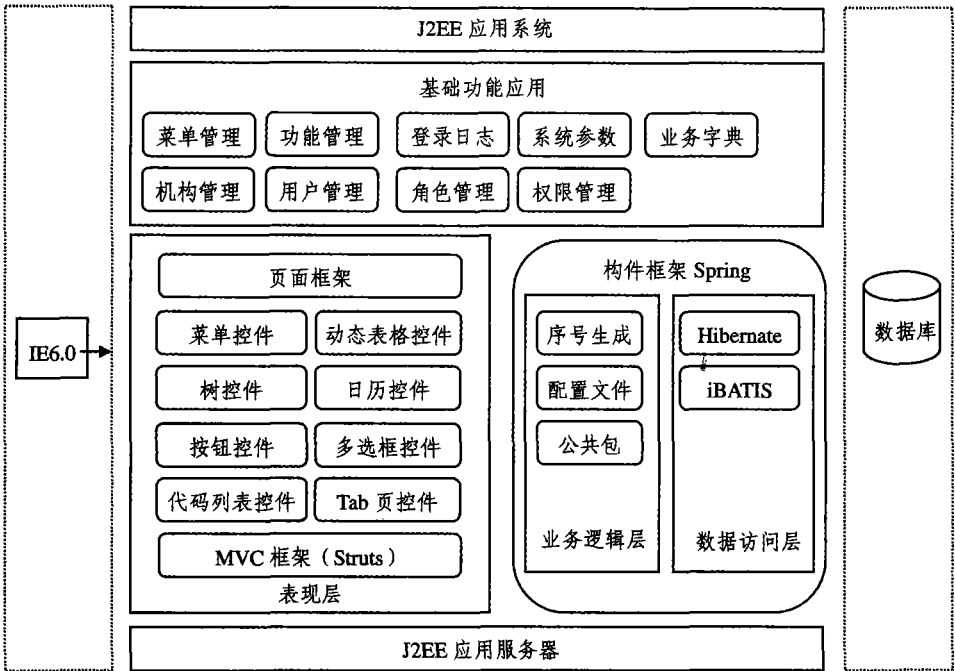


图 1 开发框架的体系结构

### 2.1 客户端

开发框架的客户端支持 IE 6.0 以上版本。

### 2.2 表现层

#### 2.2.1 MVC 框架

MVC( Model - View - Controller) 模式<sup>[10,11]</sup>把涉及数据管理和显示的功能分散到不同的对象上, 降低对象间的耦合。它把应用分成 3 部分, 分别为模型、视图和控制器, 并且尽量降低部分间的耦合。每一部分处理特定的任务, 并负责完成与其它部分的通信。Struts 是一个 MVC 框架的具体实现, 并已得到广泛的应用, 成为 Web 框架事实上的标准。本开发框架将以 Struts 作为 Web 层框架。

#### 2.2.2 页面框架

页面框架是处理 Web 界面布局和 Web 交互的统一控制的机制, 提供了多种典型的应用系统的页面框架, 以适应不同应用系统的需要。

每套页面框架包含以下内容:

- 登录界面: 用户登录界面, 可以在此基础上修改。
- 界面主框架: 包含菜单(如树形菜单、下拉菜单、Out look 菜单等)、布局(如上下结构、左右结构等)、模块界面切换管理。
- 模块界面框架: 含模块主界面和模块操作界面。
- 样式库: 样式文件, 用于控制网页的样式。
- 引用库: 包含文件, 这些公共文件在页面开发时被包含进去, 减少了页面开发时所写代码的数量和难度。使用这些包含文件, 结合框架脚本库, 基本上开发页面时只需要关注业务应用的界面元素, 和框架之间的交互一般不需要再写代

码, 如使用 Cool Button 提交页面, 提交页面后的操作成功或错误消息的提示等。

· 脚本库: 页面框架基础脚本库, 提供该页面框架下的公共脚本。

使用页面框架, 可迅速地建立应用项目的界面原型, 为下一步的开发提供直观可靠的基础。页面框架的主要特点是有统一的页面布局和操作方式, 并使用 Tab 页的方式提供页面缓存和切换。一个使用页面框架开发的程序的界面如图 2 所示。

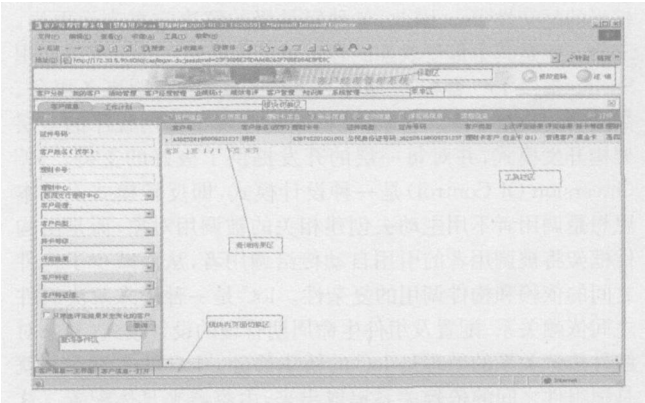


图 2 界面示例

系统界面分成系统标题区、菜单区、模块切换区、工具栏区、查询条件区、查询结果区、模块内页面切换区和信息编辑页面。其中, 系统标题区、菜单区、模块切换区、工具栏区、模块内页面切换区由开发框架统一提供, 开发应用时只需要提

供页面框架文件及相应的工具栏文件, 查询条件文件, 查询结果文件和编辑页面文件。 页面框架的脚本库提供了公共的界面交互的控制。

2.2.3 Web 控件

Web 控件是可复用的页面元素, 使用它可以加快 Web 开发的速度。 Web 控件主要包括自定义标签库、 Struts 标签库、 JSTL 标签库等。

自定义标签库娴熟运用了 JSP 定制 Tag、 DH TML、 Javas cript 等技术, 实现了应用开发时一些常用的页面控件, 这些控件使得页面具有丰富的展示功能, 包括以下控件(表 1)。

表 1

控件	说明
菜单控件 ( Menu)	动态生成多级下拉式菜单。
树控件 ( Tree)	根据具有树形结构的数据动态生成树, 支持带 radio box 和 check box 的树。
页控件 ( Tab Page)	页控件脚本, 支持 Tag 页的运行时创建、打开、查找、关闭及释放。 支持 xp, win2000 等多种样式的显示, 同时还支持带图片个性页的显示。
按钮控件 ( Cool Button)	CoolButton 类型的按钮, 支持和功能权限结合, 控制用户的功能级操作权限。
动态表格 控件 ( Dynamic Table)	支持运行时创建、删除表格行、支持键盘操作、支持页排序等。 支持列和任何控件结合, 修改单元格内容, 并且只提交变化过的数据到服务器端。
日历控件 ( Calendar)	绑定日期输入控件到日期输入域, 支持在日期输入域输入日期或在弹出选择窗口中选择日历。 支持日期输入有效性校验。 支持 Struts Form Bean 数据绑定。
多选控件 ( MultiCheck Box)	传入一个用分隔符分隔的字符串, 自动生成多个选中值的 check box。
代码列表 控件 ( CodeList)	动态生成包含值和对应的文本的 HTML OPTION 元素, 可以和 SELECT 元素结合使用。

2.3 业务逻辑层

EJB 对于不需要提供分布式服务的业务对象来说是一个重量级的模型。 对于大多数的 Web 应用来说, 分布式业务对象不是必需的, 因为 EJB 为这些类型的应用不恰当地增加了相当的复杂性<sup>[2]</sup>。 因此, 各种轻量级的解决方案应运而生。 Spring 就是其中最受欢迎的轻量级应用框架。 开发框架利用 Spring 提供 IoC 容器<sup>[13]</sup> 服务、 AOP<sup>[14]</sup> 支持、 声明事务服务、 数据访问集成服务, Spring 在框架上体现了应用程序的多层架构开发模式, 并对每一层的开发提供了极好的支持。 IoC ( Inversion Of Control) 是一种设计模式, 即反转模式。 基本思想是调用者不用主动去创建相关的被调用对象, 而是由构件框架将被调用者的引用自动传给调用者, 从而减少了构件之间的依赖和构件调用的复杂性。 IoC 是一种用来解决组件之间依赖关系、 配置及组件生命周期管理的设计模式, 其中对组件依赖关系的处理是 IoC 的精华部分。 IoC 的实际意义就是把组件之间的依赖关系提取出来, 由容器来具体配置。 这样, 各个组件之间就不存在 hard code 的关联, 任何组件都可以最大程度地得到重用。 运用了 IoC 模式后, 我们不再需要自己管理组件之间的依赖关系, 只需要声明由容器去实现这种依赖关系, 就好像把对组件之间依赖关系的控制进行了倒置, 不再由组件自己来建立这种依赖关系而交给容器去管理。 AOP( Aspect Oriented Programming) 即面向方面编程, 是面

向对象技术的延续。 它将通用的功能从不相关类之中分离出来, 同时使得这些类能通过配置的方式使用或不使用这个行为。 一旦行为发生变化, 不必修改很多类, 只要修改这个行为就可以。 利用 AOP 模式, 便于减少系统的重复代码, 降低模块间的耦合度, 并有利于未来的可操作性和可维护性。 Spring 的声明性事务管理就使用了 AOP 模式。

业务逻辑层构件采用面向接口编程, 降低了程序各部分之间的耦合。 开发框架提供了一些基础构件, 如 ID 及序列号生成、 配置文件读取、 公共工具类等。 对于某一行业的应用, 可以继续开发一些领域构件, 以支持该行业内的软件复用。

2.4 数据访问层

数据访问层将 Hibernate 和 iBATIS 两种持久化框架结合起来提供数据访问服务。 Hibernate 是一种成熟的 OR Mapping 框架, 可以大幅度减少开发时人工使用 SQL 和 JDBC 处理数据的时间, 同时具有跨数据库系统的优势。 iBATIS 对 JDBC 进行了很好的包装, 可以显著地节约数据库操作的代码量。 iBATIS 使用了一个简单的 XML 文件来配置 SQL 语句, 因此程序在不同的数据库平台上移植和修改都很方便。 对于数据的增加、 修改、 读取采用 Hibernate 操作, 而对数据的复杂查询和批量数据操作则使用 iBATIS, 因为 Hibernate 不适合在这种情况下使用, 这两者在 Spring 框架下得到了完美的集成。 数据访问层采用 DAO( Data Access Object) 设计模式<sup>[15]</sup>, 抽象和封装了所有对数据源的访问, 以便将低级别的数据访问逻辑与高级别的业务逻辑分离。 DAO 完全向客户端隐藏了数据源实现细节, 当低层数据源实现变化时, DAO 向客户端提供的接口不会变化, 所以该模式允许 DAO 调整到不同的存储模式, 而不会影响到其客户端或者业务组件。 DAO 充当了业务组件和数据源之间的适配器。

2.5 数据层

数据层一般采用关系型数据库存储应用数据。 由于采用了数据访问层隔离数据操作, 保证了开发框架可以容易地获得跨数据库的能力。 开发框架支持常用的大型关系型数据库, 如 DB2, M S S Q L Server, Oracle, Informix。

2.6 数据传输方式

关于在应用程序各层之间的数据传输方式, 开发框架有两个潜在规则。 对于记录的维护操作, 使用 Value Object 方式, 并且将 Value Object 作为 Struts Action Form 的一个属性, 实现 Value Object 和界面元素的绑定, 因此可以直接从 Action Form 中取得 Value Object。 并且, 由业务逻辑层进行业务处理后, 就可以在数据访问层通过 OR Mapping 的方式直接更新到数据库中。 整个处理过程中不需要任何数据对象之间的转换。 而且当业务数据项变化时, 只需要重新生成该值对象即可, 不需要改变相关程序的接口。 另外一个就是 Hash Map 方式, 使用 Struts 基于 Map 的 Action Form, 可以将易变的查询条件保存在 Hash Map 中, 从而在业务发生变化的时候, 业务逻辑层和数据访问层的接口甚至程序都不需要改变, 只需要修改 iBATIS 的 S Q L 配置文件即可。

2.7 统一的异常处理

开发框架对应用程序异常进行了统一的处理, 所有应用程序异常均需要继承或直接使用框架的基类异常。 应用程序抛出异常时, 不直接将错误消息文本写在代码中, 而是指定一个错误代码, 错误消息具体配置在资源文件中, 从而可以使错误消息可以配置并能够支持多语言应用。 框架的基类异常包括了 runtime exception 和 checked exception。 根据实践经验, 除非发生异常后需要程序进行干预, 否则建议使用 runtime exception, 可以减少相当多的程序代码量。 通过扩展 Struts

的异常处理框架, 开发框架支持统一的异常处理, 包括错误日志记录、错误消息在界面的显示。在异常进行统一处理后, 程序代码显得非常简洁。不是应用逻辑需要处理的异常, 均可交由一个统一的异常处理类来处理, 避免了在程序的各个部分均需要关注和处理异常。

## 2.8 基础应用功能

开发框架提供了 Web 应用系统一些通用的基础功能, 使得开发应用系统时, 只需要实现应用的业务需求即可。而对一些基础的系统功能, 则不需要进行开发或只需要进行简单的二次开发即可, 并且这些基础功能也是一个在开发框架的基础上进行应用开发的极佳范例。

基础应用功能和具体的业务应用相对独立, 主要包括: 用户管理、角色管理、权限管理、机构管理、菜单管理、系统功能管理、系统参数管理、登录日志管理、业务字典管理。

结束语 本开发框架集成了多种优秀的框架, 并提供了应用开发常用的可复用的构件、页面框架和基础应用功能。开发框架可以作为构建新的 Web 应用的模板, 在此基础上进行开发, 应用开发人员可以立刻得到一个架构良好的具有预置功能的系统框架, 极大地减少搭建系统框架的难度, 提高应用开发的效率。

## 参考文献

- 1 Szyperski C. Component Software: Beyond Object Oriented Programming. Second Edition. 北京: 电子工业出版社, 2004. 151~165
- 2 Enterprise BluePrints. <http://java.sun.com/blueprints/enterprise/index.html>
- 3 Apache Struts Project. <http://jakarta.apache.org/struts/index.html>
- 4 Hibernate project. <http://www.hibernate.org/>
- 5 Log4j project. <http://logging.apache.org/>
- 6 Johnson R. The Spring Framework Motivation and Concepts. <http://www.javapolis.com/JPO4DVContent/talks/SpringInAction1/index.html>
- 7 Gamma E, Helm R, Johnson R, et al. 设计模式. 北京: 机械工业出版社, 2000
- 8 Spring project. <http://www.springframework.org/>
- 9 iBatis project. <http://ibatis.apache.org/>
- 10 Model View Controller Pattern. <http://www.enode.com/x/markup/tutorial/mvc.html>
- 11 Stanchfield S. Advanced Model View-Controller Techniques. <http://www.ibm.com/vad.nsf/Data/Document2329>
- 12 Johnson R. xpert One on one J2EE Development Without EJB. 1-12
- 13 Fowler M. Inversion of Control Containers and the Dependency Injection pattern. <http://martinfowler.com/articles/injection.html>
- 14 JavaWorld. Aspect Oriented Programming. <http://www.javaworld.com/javaworld/jw-01-2002/jw-0118-aspect.html>
- 15 Core J2EE Patterns- Data Access Object. <http://java.sun.com/blueprints/corej2eepatterns/Patterns/DataAccessObject.html>

(上接第 214 页)

根据表 3 构造 FP tree。

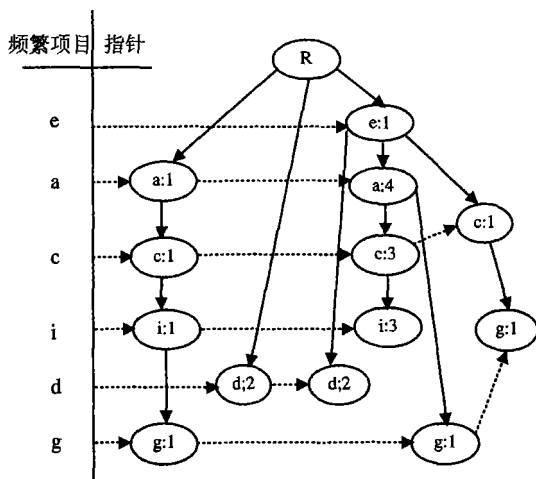


图 2 FP tree 实例

## 4.2.2 挖掘频繁项目集

FP tree 是一个压缩的数据结构, 它用较少的空间存储了后面频繁集挖掘所需要的全部信息。

FP tree 的频繁项目集挖掘是通过 FP-growth(FP tree, null)实现的。该实现过程如下:

```
Procedure FP-growth(Tree, a)
{
  if Tree 只有一条路径 P
  then 对 P 中的节点的每一个组合(记为  $\beta$ ) 做 (1)
  (1) 产生频繁集  $\beta \cup a$ , 并且把它的支持度指定为  $\beta$  中节点的最小支持度。
  else 对 Tree 的头表从表尾到表头的每一个表项(记为 a) 做 (2)
  (2) 产生频繁集  $\beta = a \cup a$ , 并且支持度为 a 的支持度
  (3) 建立  $\beta$  的条件模式库(conditional pattern base)和  $\beta$  的条件树(conditional FP tree) Tree $\beta$ 
  (4) if Tree $\beta \neq \emptyset$ 
  (5) then 调用 FP-growth(Tree $\beta$ ,  $\beta$ )
}
```

使用 FP-growth 算法, 可以得到数据库 D 的频繁项目集  $\{e\}:7, \{a\}:5, \{c\}:5, \{i\}:4, \{d\}:3, \{g\}:3, \{a, c\}:4, \{a, e\}:4, \{a, g\}:4, \{a, i\}:4, \{c, e\}:4, \{c, g\}:2, \{c, i\}:4, \{e, g\}:2, \{e, i\}:3, \{a, g, e\}:3, \{a, g, i\}:4, \{a, e, i\}:3, \{c, g, e\}:3, \{a, c, e, i\}:3\}$ 。

## 4.2.3 发现关联规则

根据 4.2.2 得到的频繁集产生关联规则。判断规则如下:

(1) 对集合 F 中的任意两个元素 A 和 B, 做判断:

if  $A \cap B = \emptyset$  并且  $A \cup B$  也是一个频繁集(即  $A \cup B$  也在 F 中)

then 计算:  $\text{count}(A \cup B) / \text{count}(A)$ 。

(2) if  $\text{count}(A \cup B) / \text{count}(A)$  不小于 min-confidence

then 产生关联规则  $A \Rightarrow B$  并存储。

(3) if F 中所有元素被遍历

then 输出所有关联规则

else 回到(1)继续执行。

这里设最小可信度为 80%, 可以得到的关联规则是: (1)

$a \Rightarrow e$  (2)  $a \Rightarrow g$ ; (3)  $a \Rightarrow i$ ; (4)  $c \Rightarrow e$ ; (5)  $c \Rightarrow I$ ; (6)  $a \Rightarrow (g, i)$ ; (7)  $(a, c) \Rightarrow I$ ; (8)  $(a, c, e) \Rightarrow i$ 。

从关联规则中可以分析出各个页面之间的关联程度, 从而了解到用户在访问网页时的习惯以及兴趣。而网站也可以根据这些分析数据对网站结构做出合理的调整, 甚至可以在用户分类的情况之下对不同用户组的访问页面有不同的推荐项目。

结论 本文在 Web 使用挖掘的数据源基础上, 通过分析 Web 服务器日志记录, 运用了基于 FP tree 的关联规则数据挖掘算法对用户行为进行分析, 为优化网站建设提供了可靠数据。

## 参考文献

- 1 谢凌, 陈新度, 陈新. 基于产品 3D 模型点击流的客户行为分析. 计算机应用, 2005, 25(12)
- 2 Kimball R, Merz R. Web 数据仓库构建指南. 北京: 清华大学出版社, 2005
- 3 Sweiger M, Madsen M R, Langston J, Lombard H. 点击流数据仓库. 北京: 电子工业出版社, 2004
- 4 夏火松. 数据仓库与数据挖掘技术. 北京: 科学出版社, 2004
- 5 陈文伟, 黄金才. 数据仓库与数据挖掘. 北京: 人民邮电出版社, 2004