



论文题目：网上购书网站

班级：_____

学号：_____

姓名：_____

目录

1. 需求分析.....	1
1.1 系统的规格说明.....	1
1.1.1 概述.....	1
1.1.2 系统功能要求.....	1
1.1.3 系统性能要求.....	2
1.1.4 系统运行环境.....	2
1.1.5 将来可能提出的要求.....	2
1.2 研究该网站所要解决的问题.....	2
2. 总体设计.....	3
2.1 系统设计.....	3
2.1.1 系统的工作流程.....	3
2.1.2 网站总体结构.....	3
2.1.3 数据流图.....	4
2.1.4 IPO 图.....	4
2.2 数据库设计.....	6
2.2.1 E-R 图.....	6
2.2.2 数据字典.....	6
2.3 选择语言.....	8
2.3.1 为什么选择 ASP.....	8
2.3.2 为什么 c#编写脚本.....	9
2.4 选择数据库.....	10
2.4.1 SQL 语言简介.....	10
2.4.2 为什么选择 SQL Server 2000.....	10
2.4.3 ADO. NET 简介.....	11
2.4.4 ADO.NET 结构.....	12
3. 详细设计与编码.....	15
3.1 主要界面设计与编码.....	15
3.2 对数据库的连接.....	23
3.3 技术难点.....	28
4. 性能测试与分析.....	29

结论.....	30
致谢.....	31
主要参考文献.....	32
外文资料翻译.....	33
中文翻译.....	33
英文原文.....	38

1. 需求分析

1.1 系统的规格说明

1.1.1 概述



上书店主要由 3 个部分组成，分别为：网站主页，顾客专区和管理员专区。网站主页是网站的门户，提供用户登录，查看新书、站内搜索，注册用户等相关功能；顾客专区是网站的主要功能模块，它主要完成顾客对图书的浏览及挑选及购物的操作；管理员专区是管理员进行后台管理，它是该网站功能最强大的平台，它主要是完成系统维护，图书的增加和删除，用户的修改及删除，销售的统计等操作的场所。

1.1.2 系统功能要求

- 顾客部分

- ✍ 选书：顾客可以根据自己的需求挑选自己所需要的书籍，并将自己需要的书放进自己的购物车里。

- ✍ 退书：在购物车中的图书如果没有点击购买，顾客可根据自己的需要进行删除。

- ✍ 查询功能：每个顾客都可以看都自己所选择的图书。顾客可以根据自己的需求搜索自己所需要的图书，该搜索可以是一个书名，一个书号，isbn 或者其他相关图书信息。

- 管理员部分

- ✍ 搜索管理：和顾客一样，拥有强大的站内搜索功能。

✍ 修改管理：对图书的信息进行修改，并可添加新书，对顾客信息进行修改和删除。

✍ 统计功能：可以按时间顺序对所销售的图书进行展示。

1.1.3 系统性能要求

本系统面向所有互联网上的人群，网站的访问量大，因此系统的可靠性要高，数据的处理能力要强。

1.1.4 系统运行环境

操作系统：

✍ Windows2000, WindowsXP

✍ 网络要求：拨号、宽带、局域网连接到互联网

1.1.5 将来可能提出的要求

增加站内会员的交流工具，尝试增加 bbs 等。

1.2 研究该网站所要解决的问题

✍ 如何设计数据库，使之不仅能满足用户对系统提出的要求，而且可靠高效。

✍ 设计界面。考虑到网站的访问量可能很大，网络压力较大，因而一个简洁实用的界面是有必要的。

✍ 如何提供一个友好的人机接口，便于顾客挑选和购买。

✍ 如何实现用户之间消息的传递。

2. 总体设计

2.1 系统设计

2.1.1 系统的工作流程

1.前期准备（数据库准备）

访客首先要登陆主页面，输入自己的 id，并注册为用户。

当注册成为户员以后，顾客就拥有购物即拥有自己的购物车，这样会员就可以选择自己所需要的书籍。

管理员登陆后直接输入自己的用户名和密码，就可以进入管理员状态对该网站进行响应的修改和操作。

2. 服务流程（系统功能开放后）

1) 会员的注册

2) 具备会员资格以后，开始挑选自己的图书。

3) 管理员在后台进行网站的维护及监管。

2.1.2 网站总体结构

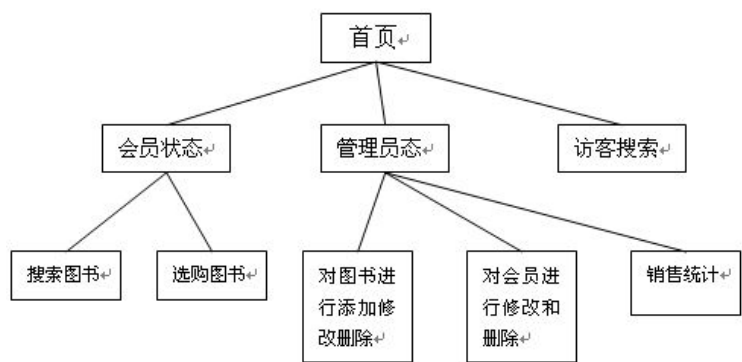


图 2.1 总体结构图

2.1.3 数据流图

顶层数据流图如下所示：

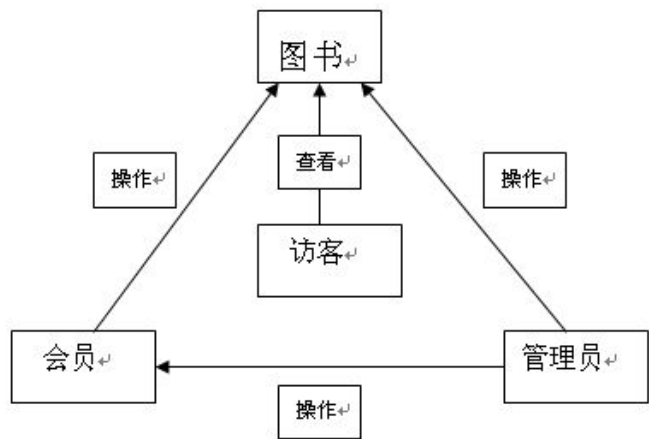


图 2.2 数据流图

2.1.4 IPO 图

下面列出系统的几个主要模块的 IOP 图：

（1）毕设题目查询模块

模块名:	图书查询模块
功能:	根据所输入的查询条件把查询结果显示出来
输入:	一个或多个查询条件
输出:	查询的结果
方法:	生成 SQL 语句, 对数据库进行查询

(2) 会员购书模块

模块名:	会员购书
功能:	根据会员的要求进行相应的判断和操作
输入:	图书编号等信息
输出:	若条件符合则选购成功, 否则操作失败
方法:	判断条件, 添加表记录

(3) 系统基本数据维护模块

模块名:	系统基本数据维护模块
功能:	主要指管理员对一些基本数据: 图书信息以及会员信息的添加、修改、删除等进行管理
输入:	系统的基本数据
输出:	输出到数据库
方法:	读写数据库

2.2 数据库设计

由于本网站需处理大量的数据,因而选择好的数据库管理系统并设计好数据库便显得异常重要了。有了好的数据库,可便于进行查询,插入删除等数据操作,提高系统的效率,也便于管理。我们选择了 Microsoft 公司的 SQL Server 2000 作为数据库管理系统,数据库设计如下:

2.2.1 E-R 图

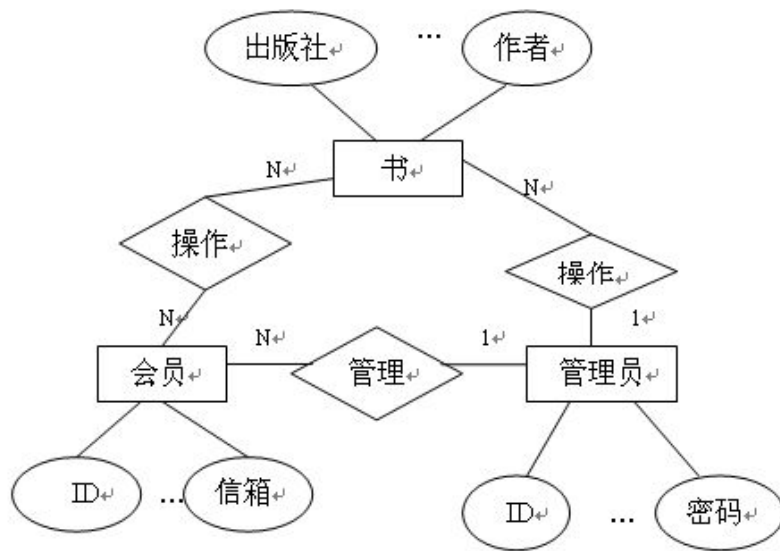


图 2.3 E-R 图

2.2.2 数据字典

1 图书表(book)

表 2.1 图书表

字段名称	类型	长度	必填字段	说明
Bno(key)	int	15	是	图书编号
Bname	varchar	40	是	书名
Bpublic	varchar	30	是	出版社
B rname	varchar	10	是	作者
Bisbn	varchar	20	是	Isbn

2 会员表(G)

表 2.2 会员表

字段名称	类型	长度	必填字段	说明
Gno(key)	Int	15	是	会员编号
Gpsw	varchar	16	是	密码
Gname	varchar	10	是	姓名
Gmail	varchar	50	否	电子信箱

3 会员—管理员表(GA)

表 2.3 会员—管理员表

字段名称	类型	长度	必填字段	说明
Gno(key)	Int	15	是	会员编号
Ano	Int	15	是	管理员编号

4 会员—图书表(GB)

表 2.4 会员—图书表

字段名称	类型	长度	必填字段	说明
Gno(key)	Int	15	是	会员编号
Bubno	Int	15	是	图书编号

5 管理员表(Administrator)

表 2.5 管理员表

字段名称	类型	长度	必填字段	说明
Ano(key)	Int	15	是	管理员编号
Apsw	varchar	16	是	密码
Aname	varchar	10	是	名字

2.3 选择语言

2.3.1 为什么选择 ASP

ASP 即 Microsoft Active Server Pages 的简称。

ASP 是一套微软开发的服务器端运行的脚本平台, ASP 内含于 IIS 当中, 目前 IIS 最高版本为 5.0 版本。通过 ASP 我们可以结合 HTML 网页、ASP 指令和 ActiveX 元件建立动态、交互且高效的 WEB 服务器应用程序。同时, asp 也支持 VbScript 和 JavaScript 等脚本语言, 默认为 VbScript。ASP 是经过服务器解析之后再向浏览器返回数据, 所以有了 ASP 就不必担心客户的浏览器是否能运行你所编写的代码。因为所有的程序都将在服务器端执行, 包括所有嵌在普通 HTML 中的脚本程序。当程序执行完毕后, 服务器仅将执行的结果返回给客户浏览器, 这样也就减轻了客户端浏览器的负担, 大大提高了交互的速度。但是这样也导致一个问题, 运行 ASP 页面相对于普通的 html 页面要慢一点, 因为普通的 html 页面只需要浏览器就能够解析, 而 ASP 则必须是服务器将整页的代码都执行一篇之后再发送数据。对于 ASP 的优点特点, 我就

不多讲了。但是 ASP 的安全性还是要提一提，由于代码是需要经过服务器执行之后才向浏览器发送的，所以在客户端看到的只能是经过解析之后的数据，而无法获得源代码，故编写者不用担心自己的代码会被别人剽窃。ASP 要浏览必须经过服务器解析才能够浏览得到。如果是将 asp 文件上传到支持 asp 的服务器，那就能够直接浏览。但是如果是在本机上面，那就必须将自己的电脑设置成为服务器。我们单机并不需要设置成为局域网，只需要设置成为单机服务器就可以。

ASP 运行所需的环境如下：

- ✍ Microsoft Internet Information Server version 3.0/4.0 on Windows NT Server
- ✍ Microsoft Peer Web Services Version 3.0 on Windows NT Workstation
- ✍ Microsoft Personal Web Server on Windows 95/98

也就是说我们经常所使用的 window95/98/2000/XP 都可以运行 asp，只需要将其设置为单机服务器就行了。

2.3.2 为什么 c#编写脚本

选择语言的时候，自己决定在诸多的编程语言中最新的、开发速度最快的语言。我们最终选择了 C#，C#是开发分布式应用尤其是基于 Web 应用的理想工具。C#的优点如下：

互用性(Interoperability)——在变成语言和开发工具中 很容易使用 C#。它也很容易和其他变成语言如 Visual Basic 和 Visual C++继承在一起。版本支持使得 C#可以兼容在它之前市场上已发行的语言。

简易性(Simplicity)——C#简单但功能强大。C#是强类型语言，具有语法智能。C#提供了 Visual Basic 的简易性，同时还与非常好的调试器和应用工具。

C#中的各种命令容易掌握使用，因为它们是它的父语言 C++和 C 的混合，大多数程序员都很熟悉。C#既有 Visual Basic 的简易性又有 C++和 C 的强大功能。

2.4 选择数据库

2.4.1 SQL 语言简介

SQL 全称是“结构化查询语言(StructuredQueryLanguage)”，最早的是 IBM 的圣约瑟研究实验室为其关系数据库管理系统 SYSTEMR 开发的一种查询语言，它的前身是 SQUARE 语言。SQL 语言结构简洁，功能强大，简单易学，所以自从 IBM 公司 1981 年推出以来，SQL 语言，得到了广泛的应用。如今无论是像 Oracle,Sybase,Informix,SQLserver 这些大型的数据库管理系统，还是像 VisualFoxpro,PowerBuilder 这些微机上常用的数据库开发系统，都支持 SQL 语言作为查询语言。

StructuredQueryLanguage 包含 4 个部分：

- ✍ 数据查询语言 DQL-DataQueryLanguageSELECT
- ✍ 数据操纵语言 DML-DataManipulationLanguage INSERT、UPDATE、DELETE
- ✍ 数据定义语言 DDL-DataDefinitionLanguageCREATE,ALTER,DROP
- ✍ 数据控制语言 DCL-DataControlLanguageCOMMITWORK、ROLLBACK WORK

2.4.2 为什么选择 SQL Server 2000

SQL Server 2000 是在 SQL Server 7.0 的基础上扩展升级的产品。SQL Server 7.0 是 Microsoft 公司在数据库领域中非常重要的产品，它具有非常好的

再造和重写特性。面市后，SQL Server 7.0 以其良好的性能、稳定性、便于管理和易于开发等优势，赢得了许多客户的喜爱，成为一个热门的数据库产品。

SQL Server 2000 继承了数据库软件设计和引擎方面的优势，已经成为新一代网络应用程序的最佳数据库和分析解决方案之一。作为 Microsoft.NET Enterprise Server 的核心组件，它将减少电子商务、在线商务和数据仓储应用程序推向市场的时间，并对需要的环境提供可靠的支持。SQL Server 2000 对 XML 和 HTTP 提供充分的支持；提供分割装载和确保时间的特性；提供先进的管理和协调机制，可以自动管理线程任务，降低总体开销。另外，SQL Server 2000 充分利用了 Windows 2000 的功能，包括支持活动目录（Active Directory）服务，最多支持 32 个处理器和 64 GB 的内存。

2.4.3 ADO.NET 简介

ADO.NET 介绍

ADO.NET 是 .NET 应用程序的数据访问模型。它能用于访问关系型数据库系统，如 SQL Server 2000，及很多其它已经配备了 OLE DB 供应器的数据源。在某种程度上，ADO.NET 代表了最新版本的 ADO 技术。然而，ADO.NET 引入了一些重大变化和革新，它们专门用于结构松散的、本质非链接的 Web 应用程序。关于 ADO 与 ADO.NET 的比较，见 MSDN 中的“用于 ADO 程序员的 ADO.NET”一文。ADO.NET 引入的一个重要变化是，用 DataTable, DataSet, DataAdapter, 和 DataReader 对象的组合代替了 ADO Recordset 对象。DataTable 表示来自一个表的行集合，在这方面它与 Recordset 类似。DataSet 表示 DataTable 对象的集合，及与其它表绑定在一起的关系和限制。实际上，DataSet 是具有内置的扩展标记语言（XML）支持的内存中的关联结构。DataSet 的一个主要特点是，它对底层的数据源一无所知，而这些数据源可能用于对其进行

填充。这是一个分离的用于表示数据集合的独立实体，并且它可通过多层应用程序的不同层由一个组件传递到另一组件。它也可作为 XML 数据流被序列化，因而非常适合于不同类型平台间的数据传输。ADO.NET 使用 `DataAdapter` 对象为发送到和来自 `DataSet` 及底层数据源的数据建立通道。`DataAdapter` 对象还支持增强的批更新特性，以前这是 `Recorder` 的相关功能。

2.4.4 ADO.NET 结构

ADO.NET `DataSet` 是 ADO.NET 的断开式结构的核心组件。`DataSet` 的设计目的很明确：为了实现独立于任何数据源的数据访问。因此，它可以用于多种不同的数据源，用于 XML 数据，或用于管理应用程序本地的数据。`DataSet` 包含一个或多个 `DataTable` 对象的集合，这些对象由数据行和数据列以及主键、外键、约束和有关 `DataTable` 对象中数据的关系信息组成。

1..ADO.NET DataSet

`DataSet` 对象是支持 ADO.NET 的断开式、分布式数据方案的核心对象。`DataSet` 是数据的内存驻留表示形式，无论数据源是什么，它都会提供一致的关系编程模型。它可以用于多个不同的数据源，用于 XML 数据，或用于管理应用程序本地的数据。`DataSet` 表示包括相关表、约束和表间关系在内的整个数据集。

2.创建和使用 DataSet

ADO.NET `DataSet` 是数据的一种内存驻留表示形式，无论它包含的数据来自什么数据源，它都会提供一致的关系编程模型。一个 `DataSet` 表示整个数据集，其中包含对数据进行包含、排序和约束的表以及表间的关系。

使用 `DataSet` 的方法有若干种，这些方法可以单独应用，也可以结合应用。您可以：在 `DataSet` 中以编程方式创建 `DataTables`、`DataRelations` 和

Constraints 并使用数据填充这些表。通过 DataAdapter 用现有关系数据源中的数据表填充 DataSet。

创建 DataSet 可以通过调用 DataSet 构造函数来创建 DataSet 的实例。请指定一个可选名称参数。如果没有为 DataSet 指定名称，则该名称会设置为“NewDataSet”。也可以基于现有的 DataSet 来创建新的 DataSet。新的 DataSet 可以是：现有 DataSet 的原样副本；DataSet 的副本，它复制关系结构（即架构）但不包含现有 DataSet 中的任何数据；或 DataSet 的子集，它仅包含现有 DataSet 中已使用 GetChanges 方法修改的行。

ADO.NET 结构的另一个核心元素是 .NET 数据提供程序，其组件的设计目的相当明确：为了实现数据操作和对数据的快速、只进、只读访问。Connection 对象提供与数据源的连接。Command 对象使您能够访问用于返回数据、修改数据、运行存储过程以及发送或检索参数信息的数据库命令。DataReader 从数据源中提供高性能的数据流。最后，DataAdapter 提供连接 DataSet 对象和数据源的桥梁。DataAdapter 使用 Command 对象在数据源中执行 SQL 命令，以便将数据加载到 DataSet 中，并使对 DataSet 中数据的更改与数据源保持一致。

ADO.NET 对象分类概述

I、Connection 对象：

表示与数据源之间的连接。可通过 Connection 对象的各种不同属性指定数据源的类型、位置以及其他属性，可用它来与数据库建立连接或断开连接。Connection 对象起到渠道的作用，其他对象如 DataAdapter 和 Command 对象通过它与数据库通信。

II、Command 对象：

表示对数据库的查询、对存储过程的调用或要返回特定表内容的直接请

求。可使用 Command 对象对数据库执行任何一种查询操作。

III、DataReader 对象：

DataReader 用于以最快的数据检索并检查查询所返回的行。可使用 DataReader 对象来检查查询结果，一次查询一行，当移到下一行时，前一行的内容就会被放弃。DataReader 不支持更新操作，由 DataReader 返回的数据是只读的。由于 DataReader 对象支持最小特性集，所以它的速度非常快，而且是轻量的。

IV、Transaction 对象：

Connection 对象有一个 BeginTransaction 方法，可用来创建 Transaction 对象。该对象可用来在 Transaction 对象的生存期提交或取消对数据库所做的更改。

V、Parameter 对象：

要使用参数化的 Command 对象，可先为查询中的每个参数创建 Parameter 对象，并将它们添加到 Command 对象的 Parameters 集合中。ADO.NET 的 Parameter 对象公开一些属性和方法，可用来定义参数的数据类型和值。

VI、DataAdapter 对象：

DataAdapter 对象充当数据库和 ADO.NET 对象模型中断开连接的对象之间的桥梁。DataAdapter 对象会填充 DataSet 对象中的表，而且能读取缓存的更改并将其提交给数据库。

3. 详细设计与编码

3.1 主要界面设计与编码

如上所述，系统的用户有 2 种不同的身份，因而须为不同的用户设计不同的页面，为各用户提供与其身份相适应的功能。本网站的设计风格为简洁、清新，部分页面设计如下：

主页设计如下：



图 3.1 主界面图

它主要是通过一下语句实现的：

```
private void btn_submit_Click(object sender, System.EventArgs e)
{ //从文件 Web.config 中读取连接字符串
string strconn= ConfigurationSettings.AppSettings["dsn"];
```

```
//连接本地计算机的 BMS 数据库

SqlConnection cn= new SqlConnection (strconn);

cn.Open ();

//构造 SQL 语句，该语句在 Users 表中检查用户名和密码是否正确

string mysql= "select * from Users where Uid='"+tbx_uid.Text +"and
UPassword='"+tbx_upassword.Text +""";

//创建 Command 对象

SqlCommand cm=new SqlCommand (mysql,cn);

//执行 ExecuteReader ()方法

SqlDataReader dr=cm.ExecuteReader ();

if(dr.Read ())

{

    lbl_message.Text= "";

    //保存当前用户名及用户权限

    Session["uid"]=dr["uid"];

    Session["upower"]=dr["upower"];

    lbl_message.Text="欢迎您！ "+Session["uid"];

    if(Session["upower"].ToString ()=="1")

    { //进入管理人员界面

        Response.Redirect ("bookmanage.aspx");

    }

    else

    { //普通用户,界面不变

    }

}
```

```
}  
else  
{  
    lbl_message.Text ="对不起，您的用户名/密码不正确，请重新输入";  
}  
//关闭连接  
cn.Close()
```

相关说明：

该界面是本网站的门户，是客户和管理员登陆网站的所必须打开的网页，该网页具有强大的功能，它拥有会员注册以及管理员登陆连接，主页上的 10 本书是最新加入数据库的 10 本书，即为新书。在主页的左边有 3 个连接，分别指向高一、高二、高三。同时，主页也拥有一个简单的搜索系统，可以对站内的书籍进行简单的搜索。

管理员添加新书的页面如下：



图 3.2 添加新书

它主要是通过一下语句实现的：

```
//利用 Command 对象调用存储过程,创建添加 bookprice 表命令类型
SqlCommand cmbookpriceadd=new SqlCommand ("bookpriceadd",cn1);
//将命令类型转为存储类型
cmbookpriceadd.CommandType =CommandType.StoredProcedure ;
//添加并给参数付值
cmbookpriceadd.Parameters .Add ("@PPID",SqlDbType.Int);
cmbookpriceadd.Parameters .Add ("@PPrice",SqlDbType.Money );
cmbookpriceadd.Parameters .Add ("@PDiscount",SqlDbType.Float );

//利用 Command 对象调用存储过程,创建添加 BookMaxBIDGet 表命令类型
//该类型返回我们所要往 book, bookprice 表中添加的 BID 值
SqlCommand cmbookmaxbidget=new SqlCommand ("BookMaxBIDGet",cn2);
//将命令类型转为存储类型
cmbookmaxbidget.CommandType =CommandType.StoredProcedure ;
//添加并给参数付值
cmbookmaxbidget.Parameters .Add ("@maxbid",SqlDbType.Int);

//将 BookMaxBIDGet 过程参数输出
cmbookmaxbidget.Parameters ["@maxbid"].Direction
=ParameterDirection.Output ;
```

```
cmbookmaxbudget.ExecuteReader ();  
  
string maxbidstring=cmbookmaxbudget.Parameters  
["@maxbid"].Value.ToString();  
  
//考虑数据库为空的情况，maxbidstring 将为空  
  
int maxbid ;  
  
if(maxbidstring=="")//如果数据库为空  
{maxbid=1 ;  
}  
  
else//如果数据库有数据  
{maxbid=Convert.ToInt32(maxbidstring)+1 ; //取出表中 BID 的最大值作为  
下一纪录的 BID  
}
```

相关说明：该页面是管理员在后台进行添加新书的操作界面，他要根据图书的信息，键入到相应的表格中，直接对数据库进行了修改。

管理员对用户的管理界面：

用户管理				
查找用户名为 <input type="text"/> 的用户 <input type="button" value="开始"/>				
用户名	用户姓名	用户类型	E-Mail	
Admin1	管理员	1	admin@admin.com	编辑 删除
yw	yewan	2	yewan@163.net	编辑 删除
sunsb	sunshaobo	2	sunshaobo@sina.com	编辑 删除
leil	leil	2	leil@sina.com	编辑 删除
admin2	超级用户2	2	888888@88.88	编辑 删除
1 2				
注：用户类型：1-管理员，2-客户				
书目管理 销售统计 退出				

我们	常见问题	关于本站	购物指南	其它
----	------	------	------	----

图 3.3 用户管理

它主要是通过一下语句实现的:

```
public void BindGrid()
{ //从文件 Web.config 中读取连接字符串
string strconn= ConfigurationSettings.AppSettings["dsn"];
//连接本地计算机的 MMS 数据库
SqlConnection cn= new SqlConnection (strconn);
//创建 SqlDataAdapter 对象,调用存储过程
SqlDataAdapter da=new SqlDataAdapter ("userlist",cn);
//创建并填充 DataSet
DataSet ds=new DataSet ();
da.Fill (ds);
dgd_userlist.DataSource =ds;
dgd_userlist.DataBind ();
cn.Close ();
}

public void DataGrid_Edit(Object sender,DataGridCommandEventArgs E)
{
dgd_userlist.EditItemIndex =(int)E.Item .ItemIndex ;
BindGrid();
}

public void DataGrid_Cancel(Object sender,DataGridCommandEventArgs E)
{
dgd_userlist.EditItemIndex =-1;
```

```
BindGrid();
}

public void DataGrid_Update(Object sender,DataGridCommandEventArgs E)
{ //从文件 Web.config 中读取连接字符串
string strconn= ConfigurationSettings.AppSettings["dsn"];
//连接本地计算机的 MMS 数据库
SqlConnection cn= new SqlConnection (strconn);
cn.Open ();
SqlCommand cm=new SqlCommand ("usermodifyForAdmin",cn);
//将命令类型转为存储类型
cm.CommandType =CommandType.StoredProcedure ;
cm.Parameters .Add ("@UID",SqlDbType.VarChar);
cm.Parameters .Add (new SqlParameter ("@UPower",SqlDbType.Int));
//从 DataGrid 中取得更新内容,Cells [0]为 UID 列
string uidvalue=E.Item.Cells[1].Text .ToString ();
cm.Parameters ["@UID"].Value =uidvalue;
//从 DataGrid 中取得更新内容
string          upowervalue=((DropDownList)E.Item.FindControl
("ddl_upower")).SelectedItem.Value .ToString ();
cm.Parameters ["@UPower"].Value =Convert.ToInt16 (upowervalue);
cm.ExecuteNonQuery ();
dgd_userlist.EditItemIndex =-1;
BindGrid();
```



```
}  
  
public void DataGrid_Delete(Object sender,DataGridCommandEventArgs E)  
{  
    //从文件 Web.config 中读取连接字符串  
    string strconn= ConfigurationSettings.AppSettings["dsn"];  
    //连接本地计算机的 MMS 数据库  
    SqlConnection cn= new SqlConnection (strconn);  
    cn.Open ();  
    SqlCommand cm=new SqlCommand ("userdelete",cn);  
    cm.CommandType =CommandType.StoredProcedure ;  
    cm.Parameters .Add ("@UID",SqlDbType.VarChar );  
    //从 DataGrid 中取得更新内容  
    //Cells [0]为 UID 列  
    string uidvalue=E.Item.Cells [0].Text .ToString ();  
    cm.Parameters ["@UID"].Value =uidvalue;  
    cm.ExecuteNonQuery ();  
    dgd_userlist.EditItemIndex =-1;  
    BindGrid();  
}  
  
public void DataGrid_Page(Object sender,DataGridPageChangedEventArgs  
E)  
{  
    dgd_userlist.CurrentPageIndex =E.NewPageIndex ;  
    BindGrid();
```

```
}

private void btn_search_Click(object sender, System.EventArgs e)
{
    string varuid=tbx_uid.Text.ToString();
    Response.Redirect ("userdetail.aspx?uid="+varuid);
}
}
```

相关说明：通过该界面，管理员可以对会员进行管理，不仅可以修改会员的信息，而且，也可以对该会员进行删除以及提高会员的权限，即可以把任意会员变成超级会员，这样，会员即有同管理员相同的权限。

3.2 对数据库的连接

1、导入库文件

使用 ADO 前必须在工程的 stdafx.h 文件最后用直接引入符号 `#import` 引入 ADO 库文件，以使编译器能正确编译。代码如下：

```
#import "C:\Program Files\common files\system\ado\msado15.dll"
no_namespace rename("EOF","EndOfFile") rename("BOF","FirstOfFile")
```

ADO 类的定义是作为一种资源存储在 ADO DLL(msado15.dll) 中，在其内部称为类型库。类型库描述了自治接口，以及 C# 用的 COM vtable 接口。当使用 `#import` 指令时，在运行时 Visual C# 要从 ADO DLL 中读取这个类型库，并以此创建一组 C# 文件。这些头文件具有 .tli 和 .tlh 扩展名，读者可以在项目的目录下找到这两个文件。在 C# 序代码中调用的 ADO 类要在这些文件中定义。

程序的第三行指示 ADO 对象不使用名称空间。在有些应用程序中，由于应用程序中的对象与 ADO 中的对象之间可能会出现命名冲突，所以有必要使用名称空间。如果要使用名称空间，则可把第三行程序修改为：`rename_namespace("AdoNS")`。第四行代码将 ADO 中的 EOF(文件结束)更名为 `adoEOF`，以避免与定义了自己的 EOF 的其他库冲突。

尽管 ASP.NET ISAPI 和辅助进程是 ASP.NET 运行时结构的主要组成部分，但还有其他一些可执行文件也发挥着作用。下表列出了所有这些组件。

名称	类型	帐户
<code>aspnet_isapi.dll</code>	Win32 DLL (ISAPI 扩展)	LOCAL SYSTEM
<code>aspnet_wp.exe</code>	Win32	EXE ASPNET
<code>aspnet_filter.dll</code>	Win32 DLL (ISAPI 筛选器)	LOCAL SYSTEM
<code>aspnet_state.exe</code>	Win32 NT	Service ASPNET

`aspnet_filter.dll` 组件是一个小的 Win32 ISAPI 筛选器，用来备份 ASP.NET 应用程序的无 Cookie 会话状态。在 Windows Server 2003 中，当启用 IIS 6 进程模型时，`aspnet_filter.dll` 还将筛选出 Bin 目录中对非可执行资源的请求。

`aspnet_state.exe` 的作用对 Web 应用程序更为重要，因为它用于管理会话状态。该项服务是可选的，可以用来在 Web 应用程序内存空间之外保存会话状态数据。该可执行文件是一种 NT 服务，既可以在本地运行，也可以远程运行。当该服务被激活后，可以将 ASP.NET 应用程序配置为将所有会话信息保存在此进程的内存中。一种类似的方案是提供更为可靠的数据存储方式，不受进程回收和 ASP.NET 应用程序故障的影响。该服务在 ASPNET 本地帐户下运行，但可以使用服务控制管理器 (Service Control Manager) 接口来配置它。

另一个应该介绍的可执行文件是 `aspnet_regiis.exe`，尽管严格来讲，它并

不属于 ASP.NET 运行时结构。该实用程序可以用来配置环境，以在一台计算机上并行执行不同版本的 ASP.NET，还可用于维修 IIS 和 ASP.NET 损坏的配置。该实用程序的工作方式是更新存储在 IIS 配置数据库的根目录和子目录中的脚本映射。脚本映射是资源类型和 ASP.NET 模块之间的一种关联关系。最后，还可以使用该工具来显示已安装的 ASP.NET 版本的状态，执行其他配置操作，如授予对特定文件夹的 NTFS 权限、创建客户脚本目录。

Web Garden 模型

Web Garden 模型可以通过 `machine.config` 文件中的 `<processModel>` 部分进行配置。请注意，`<processModel>` 部分是唯一不能放在应用程序特定的 `web.config` 文件中的配置部分。这就是说，Web Garden 模式可以应用到计算机中运行的所有应用程序。但通过使用 `machine.config` 源文件中的 `<location>` 节点，可以针对各个应用程序调节计算机的设置。

`<processModel>` 部分有两个属性可以影响 Web Garden 模型，它们是 `webGarden` 和 `cpuMask`。`webGarden` 属性接受布尔值，表示是否使用了多个辅助进程（一个相关的 CPU 对应一个进程）。默认情况下，该属性的值为 `false`。`cpuMask` 属性保存一个 `DWORD` 值，该值的二进制表示为能够运行 ASP.NET 辅助进程的 CPU 提供了位屏蔽。其默认值为 `-1 (0xFFFFFFFF)`，表示可以使用所有可用的 CPU。如果 `webGarden` 属性为 `false`，则 `cpuMask` 属性的内容将被忽略。`cpuMask` 属性还为正在运行的 `aspnet_wp.exe` 的副本数设置了上限。

Web Garden 模式使得多个辅助进程可以同时运行。但是，需要注意的是所有进程都会有自己的应用程序状态、进程内会话状态、ASP.NET 缓存、静态数据以及运行应用程序所需的其他内容。启用 Web Garden 模式之后，ASP.NET ISAPI 将根据 CPU 的数量尽可能多地启动辅助进程，每个辅助进

程都是下一进程的完整克隆（每一进程都与相应的 CPU 密切相关）。为平衡工作负荷，传入的请求以单循环的方式在运行的进程之间进行划分。辅助进程就象在单处理器中一样被回收。请注意，ASP.NET 继承了操作系统中所有的 CPU 使用限制，并且不包括实现限制的自定义语义。

总之，Web Garden 模型并不适用于所有应用程序。应用程序的状态越多，其的性能损失也越多。工作数据存储在共享内存的块中，以便一个进程输入的变化可以立即被其他进程得知。但是，处理请求时，工作数据被复制到进程的上下文中。因此，各个辅助进程将处理自己的工作数据，而应用程序的状态越多，性能损失就越大。鉴于此，仔细、明智的应用程序基准测试是绝对必要的。

只有重启 IIS 后，对配置文件中 `<processModel>` 部分所做的更改才会生效。在 IIS 6 中，Web Garden 模式的参数保存在 IIS 配置数据库中，webGarden 和 cpuMask 属性被忽略。

HTTP 管道

ASP.NET ISAPI 扩展启动辅助进程后，它将传递部分命令行参数。辅助进程使用这些参数来执行加载 CLR 前需要执行的任务。传递的值包括：COM 和 DCOM 安全性所要求的身份验证等级、可以使用的命名管道的数量和 IIS 进程标识。命名管道的名称是使用 IIS 进程标识和允许的管道数随机生成的。辅助进程不接收可用管道的名称，但可以接收识别管道名称所需的信息。

COM 和 DCOM 安全性与 Microsoft? .NET Framework 有何关系？实际上，CLR 是作为 COM 对象提供的。更准确地说，CLR 本身不是由 COM 代码构成的，但是指向 CLR 的接口却是一个 COM 对象。因此，辅助进程加载 CLR 的方式与加载 COM 对象的方式相同。

当 ASPX 请求遇到 IIS 时，Web 服务器将根据选择的身份验证模型（匿名、Windows、Basic 或 Digest）来分配一个令牌。当辅助进程收到要处理的

请求时，令牌被传递到辅助进程。请求由辅助进程中的线程获取。该线程从最初获取传入请求的 IIS 线程继承身份令牌。在 `aspnet_wp.exe` 中，负责处理请求的实际帐户取决于在特殊的 ASP.NET 应用程序中是如何配置模拟的。如果模拟被禁用（默认设置），则线程将在辅助进程的帐户下运行。默认情况下，该帐户在 ASP.NET 进程模型中为 `ASPNET`，在 IIS 6 进程模型中为 `NETWORKSERVICE`。这两个帐户都是“弱”帐户，提供的功能比较有限，可以有效抵挡回复性攻击（`Revert-to-self Attack`）。（回复性攻击是指将模拟的客户端的安全性令牌回复到父进程令牌。为辅助进程分配弱帐户可以挫败此类攻击。）

高度概括起来，ASP.NET 辅助进程完成的一项主要任务就是将请求交给一系列称为的 HTTP 管道的托管对象。要激活 HTTP 管道，可以创建一个 `HttpRuntime` 类的新实例，然后调用其 `ProcessRequest` 方法。如前所述，ASP.NET 中始终只运行一个辅助进程（除非启用了 `Web Garden` 模型），该进程在独立的 `AppDomain` 中管理所有的 Web 应用程序。每个 `AppDomain` 都有自己的 `HttpRuntime` 类实例，即管道中的输入点。`HttpRuntime` 对象初始化一系列有助于实现请求的内部对象。`Helper` 对象包括缓存管理器（`Cache` 对象）和内部文件系统监视器（用于检测构成应用程序的源文件的更改）。`HttpRuntime` 为请求创建上下文，并用与请求相关的 HTTP 信息填充上下文。上下文用 `HttpContext` 类的实例来表示。

另一个在 HTTP 运行时的设置初期创建的 `Helper` 对象是文本书写器，用于包含浏览器的响应文本。文本书写器是 `HttpWriter` 类的实例，此对象对页面代码以编程方式发送的文本进行缓存。HTTP 运行时被初始化后，它将查找实现请求的应用程序对象。应用程序对象是 `HttpApplication` 类的实例，该类就是 `global.asax` 文件背后的类。`global.asax` 在编程时是可选的，但在构建

结构时是必需的。因此，如果应用程序中没有构建类，则必须使用默认对象。ASP.NET 运行时包括几个中间工厂类，可以用来查找并返回有效的 Handler 对象以处理请求。整个过程中用到的第一个工厂类是 `HttpApplicationFactory`。它的主要任务是使用 URL 信息来查找 URL 虚拟目录和汇集的 `HttpApplication` 对象之间的匹配关系。

3.3 技术难点

本系统是一个多用户，有较大量数据处理的管理系统，涉及到网络传输、数据库操作、等技术。因而，如何保证系统在访问量大、对数据库频繁操情下能够平稳、可靠地运行，这是我们考虑的一个重点，同时也是技术难点。

首先要选择一个大型数据库管理系统，作为我们的数据库管理系统。该数据库管理系统应具备数据处理能力强、能满足多用户同时访问、快速、可靠性高等特点，可以满足对大量数据的管理。在设计本系统的数据库的时候，要本着关系数据库设计的原则，注重数据库结构的严谨、数据的独立性、查询扩充的方便性以及学校的原有的系统项兼容。在系统的编程实现方面，如何设计系统的结构，使之安全性高同时兼顾效率，对网络和数据库的压力减少；如何实现模块化设计，提高程序的通用性和程序结构的优化设计，并留着一些供日后系统扩充的接口。

4.性能测试与分析

软件测试是软件开发周期中一个极其重要的阶段，测试的目的是在软件投入生产性运行之前，尽可能多地发现软件中的错误。目前软件测试仍然是保证该软件质量的关键步骤，它是对软件规格说明、设计和编码的最后复审。

软件测试在软件生命周期中横跨两个阶段。通常在编写出每个模块之后就对它做必要的测试（称为单元测试），编码和单元测试属于软件生命周期的同一个阶段。在这个阶段结束之后，是对这个系统进行的各种综合测试。

测试名称	测试进度	测试内容	测试目的	备注
模块功能测试	在编码阶段完成	各模块的功能	编码的正确性等	个人测试
数据库测试	编码完成后一周	数据库的读写	数据库的读写是否正确	同学帮忙测试
跨平台测试	编码完成后一周	在不同的平台上使用该系统	测试系统的兼容性	同学帮忙测试
综合操作测试	编码完成后一周	对系统各个功能的综合测试	找出系统的 bug	同学帮忙测试

为了提高检测出错的几率，使测试能有计划地，有条不紊地进行，我编制了测试计划。内容如下：

通过对系统进行综合的测试，我们修正了程序中的一些错误，提高了系统的可靠性。通过测试，我们验证了需求分析中对系统功能提出的要求，得出初步结论：我作的网络书店是较为成功的。

结论

毕业设计（论文）是大学生活的重要组成部分，是对一个学生大学四年学习成果的重要检验。网络书店是我第一次完整地按照软件工程的要求，从需求分析、系统设计、详细设计到编码和软件测试所开发的系统。在老师的指导下和同学们的帮助下，我通过自己的努力，成功地开发了该系统。

网上书店的创新是：实现了网上选择为商家和顾客提供了一个便捷的交易、交流空间。

通过这个课题的开发，我深刻体会到了四年来所学课程的重要性，在实践中运用了所学的知识；我感受到了软件开发的具体过程，特别是软件工程的指导作用；模块化程序设计可以提高软件开发的效率和减少维护的负担。最重要的一个体会是：一个人的能力（包括设计和编程等）是在实践中提高的。

致谢

感谢老师在身体不便情况下，不辞辛苦，反复对我的毕业设计进行技术指导和在论文协作上的大力帮助。

感谢同学在我开发系统过程中所给予的帮助。

主要参考文献

- (1) 耿祥义，张跃平. ASP 实用教程.北京：清华大学出版社，2002
- (2) 李晓哲、张晓辉、李祥胜 . SQL Server 2000 管理及应用系统开发.
北京：人民邮电出版社，2001
- (3) 萨师煊、王珊.数据库系统概论（第三版）.北京：高等教育出版社
- (4) 王军. ASP.NET 2.0 大解密.北京：清华大学出版社.2004

外文资料翻译

中文翻译

探寻和谐的音乐舒适感

摘要：

在经过几代的音乐作品采样后，诞生这些作品的规则已经被整理成一个遗传编码并且这些具有适应环境以及繁殖自身的族群可以被选择出来。作者想要探寻的適切性是关于音乐的调和的。在这片文章中我们提出某些可以证明相关定义协和或不协和的音程的结论，可以用于音乐作品的自动谱曲。

关键词：

演化音乐，细胞自动控制，遗传算法及适应性。

绪论

普通的关于人造生命模型的观点是复杂结构可以通过一批有限的简单操作重复应用来显现，而且可以产生新兴的特征（兰顿，1997）。当应用到音乐范畴时，这些模型中最主要的目的是为构成乐曲找到最适宜的方法（米兰达，1999）。这些关系，在音乐和某些人造生命模型之间，提供了一个记号语言的三角关系含义。在这个三角关系中，我们可以用的规范代码体系，即数学模型和各种表示法，通过我们选择的这个规范的代码体系，使我们得到不同的人造结果（比洛塔，2000）。

在我们要探索的音乐范畴中，有两种重要的趋势。

第一种是探寻一种在我们可以得到的人造结果上的科学的实验法，用来检查一个音乐理论是否可以被利用来写出新的乐曲。事实上，旋律配合，管弦乐以及悦耳音调的构造的工艺某些部分经常被代码化成为清楚的准则，其他的则不是。埃里克森（1982）曾经争论过：“我们需要一种非风格限度的音乐理论，适用于世界音乐，而不仅仅是欧洲的，中国的或者是印尼之类的。”这种需要在演化音乐范畴中尤其显著。作曲家习惯于探寻悦耳的音乐构造的间隔，用来寻找刚刚好的选择，能够适合一块中的特殊瞬间，或者他们依靠一些历史风格去得到一种特殊的音乐效果或是品位。这种研究是多方面的：我们要探寻数学的，心理学的和记号语言方面的音乐。

第二种方向涉及到音乐舒适感，可以给我们某种乐曲的整体特征。在演化音乐中，音乐舒适感（像视觉的，希姆斯，1993）一般都被听众直观地感受到。它反映一种美学上的判断力，通过这种判断力，在一堆音乐作品中的某些片断，都或多或少地建立在主观的，情感的或者是知觉标准的基础上，从一个听众的感受表达出来。一些作者曾经把它作为电子乐器的根源测试过它（塔卡拉，1993），产生一个独立的韵律量度（哈洛维茨，1994），包括一个“耳朵”模型（雅各布，1995），为旋律的概念建立出一个知识基础，可以用到临时的爵士独奏曲中（比尔斯，1994），训练出一种神经网络来区别出舒适感（比尔斯，1996）。

我们采用了一种遗传算法来研究协和音程中的音乐舒适感。我们做出了一个音乐作品的样品。产生这些作品的规则已经被代码化成了遗传代码，并且我们选择了一些种群，他们有适应环境以及繁殖自身（舒适性）的能力（米切尔，1996）。我们研究的舒适感是关于协和音程的。事实上，我们在一段乐曲（暂时抛开关于音调和音调的乐曲的讨论）中的两个音符之间，评估和谐或者不和谐。某些因子随机变换产生的新群体，通过它们的适切性被选择出来，并且这个过程被重复许多次。这样繁殖许多代以后，就有可能观测出某些代的适切性，

以及乐曲结构中突然产生的特征。在这片文章中我们提出某些结论对于定义协和或不协和的音程有重要意义，可以用于音乐作品的自动谱曲。

和谐与不和谐的对立

声音传达的是哪种类型的信息?人类精神的声学的表达方式，可以用单词不能表达的的方式来描述。声音与世上自然的声音以及精神体验涉及的不同，所以和谐与不和谐就涉及到物理数学方法、心理学对声音感知的实验以及音乐声音的结构。音乐中的和谐与不和谐的问题最早是在公元前 5 世纪被毕达哥拉斯发现。他们发现，当形成音长的比率是小整数时，比如 2: 1（八度音程：两个音程之间有八个音度的间隔，其中一个每秒的振动是另一个的两倍），3: 2（五度音程：包含五个全音阶音调如 C、D、E、F 和 G 的时间间隔）和 4: 3（第四音级：自然音阶中高于或低于一给出音调四度的音调），两个连续的音符是和谐的（意思是说它们在一起听起来流畅）。在现代物理学里，这个规则已经被用来去陈述大部分音调的混合，包括在频率之间的简单比率关系。冯·赫尔蒙茨，在 14 世纪的一个德国的生理学家和物理学家，描述了关于和谐的最能被接受的理论，这个理论谴责了在振动时能被感知的不和谐音符的粗糙程度。赫尔蒙茨论断听觉系统把复杂音调或混合音调处理为独立的光谱式的成分（分音）。当两个分音的频率对听觉系统而言太相近，以至于难以被区分，听起来就好像震动（强度的周期性起伏）是这两个分音的合成。对于这个理论已经被加上更多先进的理论，成为有判断力的结合，这种结合被定义为一种频率分离，其中的分音没有被听觉系统完全区分并且接着相互作用产生形如重叠或是震动。由于无调性音乐（缺乏主旋律）以及计算机音乐的理论已经开创出了一片广阔的可能性来创造音乐作品，和谐与非和谐的对立仍然是一个难题。尽管一些作者认为和谐就是内涵的缺乏（Schoenberg, 1984），一些实验表明对于和谐的成熟的判断是适当的统一（Huron, 1991; Shellemborg and Trehub,

1994)。

怎样把一个听觉事件转化成它的一个有知觉的表达方式？当听众对于听到的东西产生一种精神上的表现，他们必须使用某种规则知道什么会发生。分组的完全形态（格式塔原理：物理的、生物的、心理的或象征的结构或形态，其构成因素并不是各组成部分间的简单相加，而是一种完整的结构或形式）在二十世纪初期就被一批德国心里学者提到，来解释为什么知觉体验的要素似乎相互间高度联系。格式塔（完全形态）这个词意思是“模式”，并且这个理论描述了大脑怎样通过形成要素间的联系来产生精神的模式。完全形态理论家论断在要素的“吸引的力量”之间一直存在有竞争，以便于从这种斗争中突然产生出的知觉构成，能成为整个知觉“范畴”的力量分配的结果，而不是成为独立部分孤立出来的属性的结果。完全形态心理学家的观点是，形成知觉构成的趋势是先天的并且自动发生的，无论何时我们感受到什么。他们提出，脱离它们整体化的形成，去觉察感官的要素是不可能的。他们论断这种结构化趋势是大脑的一种自动趋势。事实上，现在的某些研究表明对于和谐，存在一种生物学的基础，并且明确地表明和谐与不和谐是频率结合的产物。在听觉神经网络中，频率产生不同的激动震荡结构（Zentner and Kagan, 1996,1998）。

音乐的原始构成的任务是什么？音乐建立了声音的精细结构，但是，正如完全形态心理学家指出的那样，对它的美学的和知觉的理解不是单个声音的未加工的属性。它同样建立了结构的概念比如比例，模型和关键联系以及各种变换，包括运输和循环。有经验的听众利用大量的音乐概念或计划来听音乐。传统上说，音乐是从水平或垂直方位的思考。这个源自音乐符号，其中水平方位代表时间，即声音的连续形成了乐曲，而垂直方位描述定调关系，即同时发生的声音形成协调。通常，音乐家谈到音乐结构都涉及到这些类型的线索是怎样连到一起的，垂直方位与水平方位比较时有多强烈，水平方位怎样随着拍子

改变，这之类的。

由于我们想要检验音乐段落中的人工生命模型里某些突然产生的属性，我们选择把和谐作为某种意义上的舒适感来研究。我们是在假设之后开始的，假设演化在音乐里被呈现出来并且有可能对这种演化的突然产生的属性以及更高状态的结构进行探测，正如我们实际知道的那样，从某种方式来看就像是历史的演化产生了音乐体系。演化音乐里会产生什么呢？

英文原文

In search of musical fitness on consonance

Eleonora Bilotta (eleb@abramo.it), Pietro Pantano (piepa@abramo.it)

Centro Interdipartimentale della Comunicazione

Università degli Studi della Calabria,

87036, Arcavacata di Rende (CS), Italy

Abstract.

After the generation of a sample of musical compositions, the rules which gave birth to these groups of compositions have been codified into a genetic code and these families, which had the capability of adapting to the environment and to reproduce themselves have been selected. The fitness the authors were in search for was about musical consonance.

In this paper we present some results which could prove interesting in defining consonance/dissonance, operated automatically on musical compositions.

Key words:

Evolutionary Music, Cellular Automata, Genetic Algorithms and Fitness.

Introduction

The common thesis in Artificial Life models is the idea that complex structure can emerge from the repeated application of a limited set of simple operations and give birth to emergent properties (Langton, 1997). When applied in a musical context, one of the major goal of these models is to find the optimum method for structuring musical compositions (Miranda, 1999). This relationship, between

music and some AL models, furnishes a semiotic triangle of signification (or musification). In this triangle, mathematical structures, the codification system we can use and various kinds of representations let us obtain artificial artefacts of a different kind, according to the codification systems we have chosen (Bilotta et al., 2000). There are two important directions in which we are going in exploring the musical context. The first is a scientific experimentation on the kind of artefacts we can obtain in order to detect a musical theory that can be utilised to write new kind of compositions. In fact, while some parts of the crafts of counterpoint, orchestration and construction of melodies are oftencodified into explicit rules, others are not. Erickson (1982) has argued that: "We need a music theory that is not style bound, that is valid for world music, not its European, Chinese or Indonesia varieties". This lack is especially evident in the context of evolutionary music. Composers are used to search the space of musical constructs to find just the right choice for a particular moment in a piece or they rely on some historical style in order to obtain a particular musical effect or taste. This research is multifold: we are exploring the mathematical, psychological and semiotic aspects of music. The second direction concerns musical fitness, which can give us some global characteristics of a melody. In evolutionary music, musical fitness (like visual one, Sims, 1993) is generally operated intuitively by listeners. It reflects an aesthetic judgement according to which some pieces, in a population of compositions, are better or worse, based on subjective, emotional or perceptual criteria, expressed by a listener's sample. Some authors have detected it as the source for a digital instrument (Takala, 1993), generating a single rhythm measure (Horowitz, 1994), evolving an "ear" module (Jacob, 1995), building up a knowledge base of melodic

ideas for use in improvising jazz solos (Biles, 1994), training a neural net to discriminate fitness (Biles et al., 1996). We have used a genetic algorithm to investigate musical fitness on consonance. We have generated a sample of musical compositions. The rules which give birth to these groups of compositions have been codified into a genetic code and we have selected those families, which had the capability of adapting to the environment and of reproducing themselves(fitness) (Mitchell, 1996). The fitness we were in search of was about musical consonance. In fact, we computed the consonance and dissonance values between two notes in a melody(forgetting for a moment the discussion between tonal and a-tonal music). The offspring generations, created by some genes random mutation, have been selected according to their fitness and the process has been repeated many times. After many generations, it has been possible to observe an empowerment of some populations' fitness and of emergent properties in melodic organisations. In this paper we present some results which could be of interest in defining consonance/dissonance, operated automatically on musical compositions.

The opposition consonance/dissonance

What sort of information does sound convey? Human mental representations of acousticevents can be elaborate in a way that the word sound doesn't offer. Sound refers indifferently to the physical sound in the world and to the mental experience of it, so that

consonance/dissonance is related to physical-mathematical theories, to psychological

experiments on sound perception and to the musical organisation of sound. The problem of consonance/dissonance in music began with the Pythagoreans in

the fifth century B. C. They discovered that two notes produced by strings were consonant (which means they sounded smooth together) when the ratios of the lengths of the strings were formed from low integers, such as 2:1 (the octave), 3:2 (the fifth) and 4:3 (the fourth). In modern Physics, the rule has been replaced to state that the most tonal combination involve simple ratio relationships between frequencies. Von Helmholtz, a German physiologist and physicist in the XIXth century, elaborated the best-accepted theory of consonance, which blames the roughness of dissonant tones on the perception of beats. Helmholtz argued that the auditory system resolves complex tones or mixture of tones into individual spectral components (partials). When two partials are too close in frequency for the auditory system to resolve them, it hears the beats (periodic fluctuations of intensity) created by their summation. To this theory has been added the more recent theory of critical band, which is defined as a certain frequency separation, within which partials are not fully resolved by the auditory system and then interact to produce phenomena such as masking or beats. The opposition consonance/dissonance is still a problem, since the theories of atonal music and computer music have opened a broad range of possibilities in creating musical compositions. While some authors consider consonance as lacking in meaning (Schoenberg, 1984), some experiments show that the adult judgement about consonance is well consolidated (Huron, 1991; Shellemborg and Trehub, 1994). How can an auditory event be translated into a perceptual representation of it? When listeners create a mental representation of the auditory input, they must employ rules about what happens. Gestalt's principles of grouping were evolved by a group of German Psychologists in

the early part of the XXth century to explain why elements in sensorial experience seemed highly connected to one another. The word Gestalt means "pattern" and the theory described how the brain created mental patterns by forming connections between the elements. Gestalt theorists argued that there was always competition between "the forces of attraction" of elements, so that the perceptual organisation that emerges from this conflict would be a consequence of the distribution of forces across the whole perceptual "field" and not of the properties of individual parts taken in isolation. The Gestalt psychologists' view was that the tendency to form perceptual organisations was innate and occurred automatically whenever we perceived anything. It was impossible, they claimed, to perceive sensory elements without their forming an organised whole. They argued that this organisational tendency was an automatic tendency of the brain. In fact, some recent research shows that there is a biological basis for consonance and that results show clearly that consonance and dissonance are combinations of frequencies which produce different stimuli configurations in the neural net disposed for the aural perception (Zentner and Kagan, 1996, 1998). What is the role of primitive organisation in music? Music builds elaborate structures of sounds, but its aesthetic and perceptual comprehension is not the raw properties of the individual sounds, as Gestalt Psychologists pointed out. It also builds upon structural concepts such as scales, model and key relations and a variety of transformations, which include transportation and repetition. Experienced listeners make use of a large number of musical concepts or schemas in listening to music. Traditionally, music is thought of as having a horizontal and a vertical dimension. This derives from musical notation in which the horizontal dimension stands for

time and the succession of sounds that forms the melody, and a vertical one which depicts pitch relations, or the simultaneous sounds that form harmonies. Usually, musicians speak of musical texture in referring to how these types of threads go together, how strong the vertical dimensions are in comparison with the horizontal ones, how the horizontal ones change over time, and so on. Since we want to verify some of the emergent properties of AL models in the musical context, we choose to work on consonance as a means of fitness. We are following the hypothesis that evolution is present in music and it is possible to detect in this evolution emergent properties and higher order organisation, which in some way resemble the historical evolution that has produced musical systems, as we actually know it. What happens in evolutionary music?