

浙江大学硕士学位论文

联合循环机组辅助决策系统的设计与实现

刘昱

2007 年 6 月

摘 要

燃气—蒸汽联合循环机组在效率 and 环境保护方面相比传统的火力发电技术都有很大程度的提高,随着技术的不断进步,我国东部地区相继建成了一批以天然气为燃料的重型联合循环电站。本文以它们中具有代表性的 9F 级单轴燃气—蒸汽联合循环机组为研究对象,设计并实现联合循环机组辅助决策系统,以提高电厂的维修和运行水平。

辅助决策系统包括维修间隔计算模块和在线性能计算模块,两个模块的数学模型具有相对独立性。维修间隔计算模型和步骤主要根据制造厂提供的重型燃气轮机维修资料确定,它把燃气轮机分为燃烧系统、热通道和转子三个部件,将外因对这三个部件的影响转化为它们的因素启动次数和因素运行时间,最后由这两个指标来确定维修间隔。

在线性能计算模型和步骤主要根据联合循环的基本原理确定,将联合循环分为压气机、燃烧室、燃气透平、余热锅炉、蒸汽轮机及联合循环机组整体,并分别计算它们的热力指标,之后通过实验曲线拟合的公式对整机关键性能参数进行修正。其中结合了国内外相关文献资料提供的一些简化算法和经验公式,使之更符合计算机计算的要求。

在系统的实现过程中使用了 .NET 程序设计框架、PI 实时/历史数据库编程接口等软件开发技术,论文详细阐述了上述技术在辅助决策中的角色和应用方法。在此基础上,论文还介绍了为提高系统的可扩展性和用户体验,而在程序开发过程中使用的目前流行的对象关系映射技术和异步 JavaScript 和 XML 技术。

文章的最后一部分阐述了辅助决策系统及其服务器端计算程序、WEB 应用程序的架构,讨论了燃气和水蒸汽热力性质计算程序、维修间隔计算程序和在线性能计算程序的具体实现方法。

关键词: 联合循环; 维修间隔; 性能计算; PI 数据库; .NET 框架

ABSTRACT

Gas-steam combined cycle power plant improves a lot on efficiency and environmental protection compared to traditional thermal technology. With the continuous progressing of technology, lots of combined cycle power plants have been built in the Eastern China. In the paper, we take uniaxial 9F grade combined cycle unit as representative to study, design and realize decision making assistant system of combined cycle, enhancing the plant maintenance and operation.

Decision making assistant system contains maintenance interval calculation module and online performance calculation module, but mathematical model of the modules is relative independent. Maintenance interval model and procedures are confirmed mainly according to heavy gas turbine maintenance data set provided by manufactory. Gas Turbine is divided into combustion, hot gas path and rotor components. The maintenance interval is defined by the rectangle established on the factor-starts and factor-hours which take factors influence the interval into consideration.

Online performance computing model and procedures are primarily based on the basic principles of combined cycle. It puts combined cycle unit into compressor, combustor, gas turbine, HRSG, steam turbine and the overall combined cycle unit and calculate thermal parameters separately for them. The system takes empirical formula to amend the key parameters. then, it combines domestic and international literature information on the simplified algorithm to reduce the amount of electronic computation.

.NET programming framework and PI real-time/historical database programming interface software technology used in the implementation of system are elaborated on the thesis of their roles and usage. In order to improve the scalability and user experience, the article also introduces currently popular programming skills such as ORM and Ajax used in the system

The final part of the article discusses the framework of combined cycle decision making assistant system, server calculation program and web application. The specific program implementation of gas and steam thermodynamic properties calculation procedures, maintenance interval computational procedures and on-line performance calculation procedures is another keystone.

Keywords: Combined Cycle; Maintenance Interval; Performance Computing; PI; .NET

目 录

第一章 绪论.....	1
1.1 辅助决策系统的开发背景.....	1
1.2 辅助决策系统的研究内容.....	2
第二章 辅助决策系统的原理和模型.....	4
2.1 维修间隔计算的原理和模型.....	4
2.1.1 维修间隔计算概述.....	4
2.1.2 热通道、燃烧系统和转子的标准维修间隔.....	5
2.1.2 热通道因素运行时间的确定.....	6
2.1.3 热通道因素启动次数的确定.....	7
2.1.4 燃烧系统因素运行时间的确定.....	7
2.1.5 燃烧系统因素启动次数的确定.....	9
2.1.6 转子因素运行时间的确定.....	10
2.1.7 转子因素启动次数的确定.....	10
2.1.8 燃气轮机设备的更换间隔.....	11
2.2 在线性能计算的原理和模型.....	13
2.2.1 在线性能计算概述.....	13
2.2.2 燃气轮机组的模型和计算方法.....	14
2.2.3 余热锅炉的模型和计算方法.....	17
2.2.4 蒸汽轮机的模型和计算方法.....	19
2.2.5 联合循环机组的性能计算与修正.....	21
第三章 程序设计框架与技术在系统中的应用.....	26
3.1 系统的开发平台—Microsoft .NET	26
3.1.1 .NET 框架概述.....	26
3.1.2 C#程序设计语言	28
3.1.3 WEB 开发与 ASP.NET	29
3.1.4 数据库与 ADO.NET	34
3.2 PI 实时/历史数据库	36
3.2.1 SIS 系统与 PI 数据库	36
3.2.2 PI-ProcessBook 与 PI-ActiveView	38
3.2.3 PI 数据库应用程序接口 PI-API.....	42
3.2.4 PI 数据库软件开发工具 PI-SDK	45
3.3 对象关系映射 ORM	48
3.4 异步 JavaScript 和 XML	52
3.4.1 Ajax 概述	52

3.4.2 ASP.NET AJAX 框架	54
第四章 辅助决策系统的架构和实现.....	58
4.1 辅助决策系统的架构.....	58
4.1.1 半山发电厂 SIS 系统.....	58
4.1.2 辅助决策系统架构模式的选择.....	58
4.1.3 服务器端计算程序架构.....	60
4.1.4 WEB 程序架构.....	60
4.1.5 系统的总体架构.....	61
4.2 燃气和水蒸汽的热力性质计算.....	61
4.2.1 燃气热力性质计算方法.....	61
4.2.2 水蒸汽热力性质计算方法.....	64
4.3 维修间隔计算的实现.....	65
4.4 在线性能计算的实现.....	68
4.5 辅助决策系统计算结果.....	69
4.6 辅助决策系统截图.....	71
4.6.1 服务器端计算程序.....	71
4.6.2 WEB 应用程序.....	72
第五章 结论与展望.....	74
参考文献.....	76
研究生期间的科研工作和发表论文.....	78
致谢.....	79

第一章 绪论

1.1 辅助决策系统的开发背景

在世界范围内,燃气轮机及其联合循环发电已成为火力发电的主要发展方向之一。二十世纪九十年代,由于燃气—蒸汽联合循环机组所具有的明显优越性能,各国特别是发达国家加强了研发力度,其技术出现了很大飞跃,国外大公司相继推出了先进的大功率高效率的燃气—蒸汽联合循环机组,如 GE、西门子、三菱等公司的单机功率超过 300MW,热效率超过 45% 的联合循环机组^[1]已投入实际运行。随着这些先进的大型联合循环机组投入商用,燃气轮机及其联合循环在世界电力系统中的地位发生了明显的变化,在世界发电容量中所占份额明显地快速增长,它们不仅可以被用于电网紧急备用电源和承担尖峰负荷,而且还能携带基本负荷和中间负荷。

在我们国家,特别是东部地区,经济发展迅速,但一次能源匮乏。目前火力发电厂以煤炭为主,环境污染日趋严重。随着我国西气东输工程的完成和配合沿海省份 LNG(液化天然气)接收站的建设,在浙江、福建和广东等省份已经建设完成和正在建设多个大型的燃气—蒸汽联合循环电站。相比普通的燃煤电站,燃气—蒸汽联合循环电站具有热效率高、排放污染物低、建设周期短、负荷适应能力好的优点,而这些优点正好弥补了当前火力发电的不足之处,所以联合循环发电也是今后我国火力发电鼓励发展和最有前途的方向之一。

随着联合循环理论和技术的发展,机组的功率和设备复杂程度都在提高,因此对机组的稳定性也有了更高的要求,另一方面,为了提高电厂在市場中的竞争力,机组的经济性也尤为重要。如果将目前迅速发展的计算机技术和网络技术应用到机组状态监测和分析中,能够帮助技术人员更好的判断和提高机组的稳定性与经济性。

国外以 GE 为代表的动力设备制造厂,在燃气轮机和联合循环热力设计以及试验方面经过多年的研究,积累了大量的经验,因而也有相应的比较通用的燃气轮机辅助决策系统软件包,软件包的内容包含了本课题研究的在线性能计算模块和维修间隔计算模块等。在软件包的基础上美国 GE 公司进一步结合了互联网的优势,推出了实时远程辅助决策系统,该系统通过互联网把分布于世界各地连入该系统的联合循环机组的运行数据传输到设在美国亚特兰大的控制中心,控制中心人员通过使用计算机程序或者专家经验分析数据,然后再将运行状况分析结果返回给联合循环机组的运行人员,以帮助他们决策,更好的使联合循环机组安全、稳定、经济的运行。据 GE 公司工作人员介绍,至今该系统已成功运营了 3 年,世界范围内已有 100 多台联合循环发电机组连入系统,并且以后将不再单独

出售辅助决策软件包,如果想继续使用GE的辅助决策系统必须通过该远程方式。GE的辅助决策系统在国外很多GE机组上已经成功运用,包括他的远程系统,实践证明这是一个可靠的和安全的联合循环机组辅助决策系统。但是基于以下考虑,我认为虽然这样一个系统是可靠的和安全的,但是我国还是不适合使用。第一,辅助决策计算软件包通常价格不菲,而且随着计算模块的增加,价格成倍增加,而且在转向远程系统以后,不单独出售软件包,而是必须通过支付年使用费来使用该系统,第一次使用还需要投入设备租赁费来安装数据采集,远程传输设备,如取消该项服务,设备归还制造厂。第二,在使用远程系统以后,它采集的数据是封闭的,不向当地的运行人员显示,只是返回分析结果,因此不知道远程系统到底采集了哪些信息。远程系统在一定权限下可以实际控制联合循环机组,因此对于电厂的安全运行,甚至国家安全造成潜在威胁。

随着燃气轮机和联合循环机组在我国逐渐得到重视和应用,我国的很多相关单位,如华北电力大学,中国科学院工程热物理所,浙江大学等在联合循环理论和计算机辅助系统方面进行了多项研究,虽然相比国外的研究成果有一定的距离,但是也取得了不小的成绩。重庆大学以我国目前使用较多的GE公司9E型联合循环机组为研究对象,根据测量透平排气温度法,通过燃气轮机排气温度 T_4 、压气机排气压力 P_2 、大气温度 T_a 等参数实现对燃气轮机透平入口温度的间接测量,克服了目前国内透平入口温度 T_3 仅仅局限于机组在基本负荷时的温度。实现了在全工况下对 T_3 的测量^[2]。浙江大学根据厂家提供的特性曲线及修正曲线,用曲线拟合的方法建立数学表达式,利用这些公式来计算联合循环变工况性能,并通过试验来校核模型,对二拖一型300MW联合循环机组变工况下运行的最佳匹配方式和机组开停机时间的最优化分配进行了研究。针对两种不同的负荷调度方式,建立了不同的优化模型,研制两种不同的负荷优化系统:在线优化系统和离线优化系统。在线优化系统根据中调的调度负荷、环境信息(大气温度、大气压力等)和建立的联合循环机组最优化负荷分配模型,完成对各机组的实际负荷分配和协调;离线优化模型则根据预测负荷与预测环境参数,并考虑机组运行经济性、安全性以及各种约束条件,确定机组在一个调度周期内的启停情况和负荷分配,使机组在本周期内所消耗的费用最少。目前该研究成果已在浙江温州龙湾联合循环电厂的机组投入实际使用,取得了良好的经济效益^[3]。

1.2 辅助决策系统的研究内容

本文以我国投产的第一套9F级燃气—蒸汽联合循环机组,杭州华电半山发电厂3台美国通用电气(GE)公司生产的9F级S109FA型燃用天然气、单轴重型燃气轮机联合循环机组为研究对象,归纳重型燃气轮机的维修间隔计算方法和模型,根据历史数据库中的数据统计维修间隔;分析研究燃气轮机、余热锅炉和

汽轮机的性能计算模型、运用燃气—蒸汽联合循环热力性能计算和变工况分析等热工理论，对联合循环发电机组进行在线性能计算。系统覆盖了联合循环的运行和维修两方面，与联合循环的安全、稳定、经济性紧密相关，因此在理论和应用研究具有重要的意义。

根据联合循环机组进行理论研究和电厂实际运行的需要，本课题的辅助决策系统包含下述三个方面的内容：

1. 联合循环机组维修间隔计算。

考虑燃料类型、最大负荷和其他因素对机组设备运行寿命的影响，计算机组的因素启动次数和因素运行时间，并根据这两个参数综合决定机组的实际维修间隔，从而使设备在安全的前提下具备最大的可用率。

2. 联合循环机组在线性能计算。

从数据库中读取机组的运行数据，根据燃气—蒸汽联合循环的基本原理计算机组的效率、热耗率和气耗率等关键运行参数，并使用修正曲线和拟合公式把它们修正到设计工况，帮助工作人员判断电厂运行状况、提高电厂的运行水平。

3. 使用 Microsoft .NET 程序设计框架、PI 客户端工具和 PI 应用程序接口等软件开发技术完整实现联合循环机组辅助决策系统。

第二章 辅助决策系统的原理和模型

因为联合循环机组辅助决策系统的各个模块的原理和模型有所不同，具有相对的独立性，因此在本章将依据每个不同的模块进行说明。

2.1 维修间隔计算的原理和模型

2.1.1 维修间隔计算概述

随着发电机组的容量的不断增大，如半山的联合循环机组额定功率已达到390MW，从而对机组和设备的可靠性提出了越来越严格的要求，因为大容量机组因事故跳闸会对电网产生很大的冲击。而对机组和设备的维修正是保证机组可靠运行的一个重要手段。判断机组是否应该维修存在两种方法，一种是按时维修即“几年一小修，几年一大修”，将机组的维修时间固定起来，到了预定的时间，无论机组运行情况如何，机组是否应该维修都要进行维修，而对于未达到预订时间却真正应该维修的机组没有维修而继续运行，从而带来了安全隐患。正是因为这种按时维修存在这样的不可靠性，越来越多的电厂将按时维修改为了按需维修，按需维修即通过对运行参数或者其他参数进行监测和分析来评定机组是否需要维修。这样一种方式让机组在具备安全保证的情况下具有最大的可用率，从而使效益最大化。

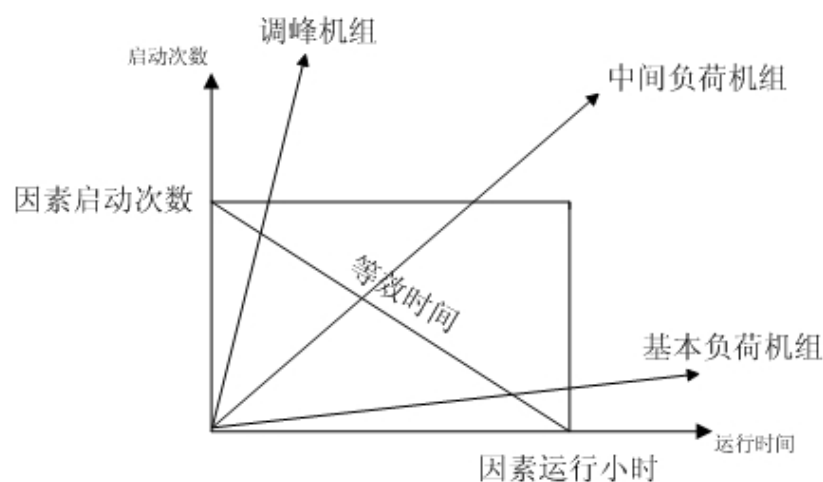


图 2.1 维修间隔的两种确定方法

联合循环维修间隔的计算与它的运行状况如燃料类型、环境和负荷特点等存在联系。目前的联合循环维修间隔计算存在两种计算方法^[4]，一种是把外因的影响转化为因素启动次数和因素运行时间，由这两个指标来确定维修的时间；另外一种是把外因的影响和每一次启动循环转化为等效运行时间，通过等效运行时间

这一个指标决定维修间隔。两者的差异如图 2.1 所示，第一种方法的检查间隔期由启动次数和运行时间围成的矩形所确定，而等效运行时间检查的检查间隔期由一根从启动次数极限点到运行时间极限点的直线围成。

半山发电厂的机组提供商 GE 认为采用第一种方法能够在保证可靠性的情况下得到最大的可用率，特别是对于中间负荷机组，因此在半山发电厂联合循环机组辅助决策系统的维修间隔计算将采用因素启动次数和因素运行时间共同决定间隔时间的方法。

维修间隔的长短与联合循环机组的燃料类型、环境和负荷特点等存在密切的联系。联合循环机组的供应商和制造厂在以天然气为燃料，基本负荷运行和没有注水或者蒸汽等基准条件下先给出了一个最长的维修间隔周期，对于不同于此基准的运行，根据每次运行的实际情况计算出一个维修系数，并通过这个维修系数和理想情况下的维修间隔周期决定因素启动次数和因素运行时间，最后根据这两个参数来决定实际的维修周期。

2.1.2 热通道、燃烧系统和转子的标准维修间隔

除了环境和运行参数对维修间隔的影响，燃气轮机的不同部件的维修间隔也不相同，比如那些与燃烧过程有联系的、暴露在从燃烧系统排出的高温烟气中的热通道部件如火焰筒、透平喷嘴等，他们的维修周期就肯定比燃气轮机的转子的维修周期短。因为燃气轮机的部件非常多，不可能为每个部件都设置一个维修周期，所以只能选择几个具有代表性的部分来设置它们的维修间隔，其他的根据各自的情况与这几个计算值联系起来，比如说火焰筒的维修间隔为一个热通道维修间隔，而它的更换间隔设定为五个热通道维修间隔。

GE 公司的技术人员经过研究和实际经验的积累，选择了热通道、燃烧系统和转子这三个具有代表性的部件做为基准并得出了 GE 公司生产的各型燃气轮机在以天然气为燃料、基本负荷运行、没有注水和蒸汽以及有和无 DLN 燃烧的理想条件下运行时，燃烧系统、热通道和整机的用运行时间和启动次数决定的理想检查间隔期，如表 2.1。

表 2.1 GE 燃气轮机在标准情况下的理想维修间隔

检查形式	燃烧室形式	运行时间(小时)/起停次数(次)				
		MS6B	MS9E	MS6FA	MS9F/FA/FA+	MS9FA+e
燃烧系统	无 DLN	12,000/1,200	8,000/900	—	—	—
	DLN	12,000/450	12,000/450	8,000/450	8,000/450	8,000/450
热通道		24,000/1,200	24,000/900	24,000/900	24,000/900	24,000/900
转子		48,000/2,400	48,000/2,400	48,000/2,400	48,000/2,400	48,000/2,400

在燃气轮机的实际运行中, 机组的运行条件不可能是理想的, 燃料、负荷设定、水或蒸汽喷注、尖峰负荷运行、机组跳闸、起动方式、热通道部件的设计等因素都会影响检修周期。因此, 对于表 2.1 中理想的检查间隔, 要根据运行情况, 从联合循环机组运行生产的 PI 实时数据库中, 查询各相关实时参数, 进行计算、分析、判断, 形成燃气轮机的运行状态描述, 对理想的检查间隔进行修正, 从而得出实际检查间隔。

联合循环机组的三个维修基准, 按先高温部件、后其它部件的次序分为三项, 即燃烧系统检查、热通道检查和转子检查。这三个检查是相互独立的, 即每种检查的维修间隔有自己的计算方法; 同时它们又有等级关系, 热通道检查包括燃烧系统检查, 而转子检查又包括热通道检查和燃烧系统检查, 即到了热通道检查间隔, 同时也就检查了燃烧系统, 检查后不只热通道维修间隔重新计算, 燃烧系统的维修间隔也重新开始计算。各项基准的维修间隔修正计算数学模型阐述如下。

2.1.3 热通道因素运行时间的确定

热通道因素运行时间由公式(2.1)给出

$$HGPI_{\text{hour}} = \frac{HGP_{\text{max_hour}}}{\frac{(K + M \times I) \times (G + 1.5D + A_f H + 6P)}{G + D + H + P}} \quad (2.1)$$

式中:

$HGP_{\text{max_hour}}$ — 基准情况下的热通道最大维修间隔运行时间, 从表 2.1 查出

M 和 K — 水\蒸汽注入系数, 取值见表 2.2

I — 水\蒸汽注入量与进口空气流量比

G — 燃用天然气的每年基本负荷运行时数

D — 燃用轻油的每年基本负荷运行时数

H — 燃用重油的每年基本负荷运行时数

P — 每年的调峰运行小时数

A_f — 重油严重系数, 渣油取 2-4, 原油取 2-3

表 2.2 热通道维修间隔(运行时间)的水\蒸汽注入系数

M	K	控制方式	蒸汽注射率	N2/N3 材料
0	1.0	干	<2.2%	GTD-222/FSX414
0	1.0	干	>2.2%	GTD-222
0.18	0.6	干	>2.2%	FSX414
0.18	1.0	湿	>0%	GTD-222
0.55	1.0	湿	>0%	FSX414

2.1.4 热通道因素启动次数的确定

热通道因素运行时间由公式(2.2)给出

$$HGP_{I_{start}} = \frac{HGP_{max_start}}{0.5N_A + N_B + 1.3N_P + 20E + 2F + \sum_{i=1}^{\eta} (a_{Ti} - 1)T_i} \quad (2.2)$$

$$N_A + N_B + N_P$$

式中：

- T_i — 跳闸中的第 i 次
- HGP_{max_start} — 基准情况下的热通道最大维修间隔启动次数，从表 2.1 查出
- N_A — 每年部分负荷启动/停机循环次数
- N_B — 每年基本负荷启动/停机循环次数
- N_P — 每年尖峰负荷启动/停机循环次数
- E — 每年紧急启动次数
- F — 每年快速升负荷启动次数
- η — 跳闸次数
- a_{Ti} — 跳闸严重系数，取值与跳闸负荷有关，见图 2.2

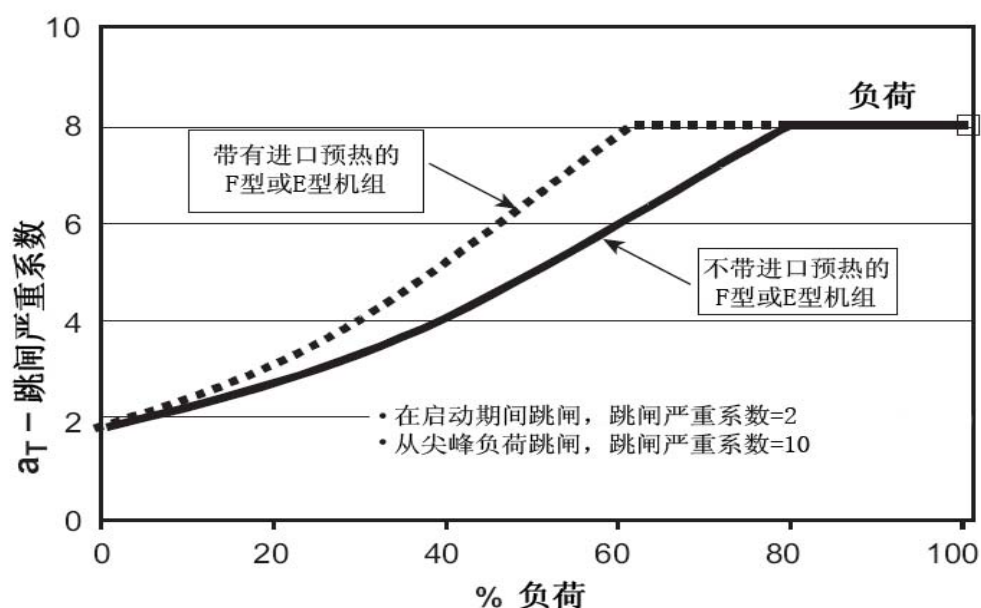


图 2.2 跳闸系数与负荷的关系

2.1.5 燃烧系统因素运行时间的确定

燃烧系统因素运行时间由公式(2.3)给出

$$CI_{\text{hour}} = \frac{CI_{\text{max_hour}}}{\frac{\sum(K_i \times A_{fi} \times A_{pi} \times t_i)}{\sum(t_i)}} \quad (2.3)$$

式中：

- i — 运行模式的第 i 种
- $CI_{\text{max_hour}}$ — 基准情况下的燃烧系统最大维修间隔运行时间
从表 2.1 查出
- K — 水\蒸汽注入系数，根据表 2.3 确定
- A_f — 燃料系数，根据表 2.4 确定
- A_p — 负荷系数，根据表 2.5 确定
- t_i — 每种模式的运行时间

表 2.3 燃烧系统维修间隔(运行时间)的水\蒸汽注入系数

运行方式	K
注入蒸汽、干控制曲线	$\text{Max}(1.0, \exp(0.34(\text{蒸汽比}-2.0\%)))$
注入蒸汽、湿控制曲线	$\text{Max}(1.0, \exp(0.34(\text{蒸汽比}-1.0\%)))$
注入水、干控制曲线	$\text{Max}(1.0, \exp(1.80(\text{水燃比}-0.80)))$
注入水、湿控制曲线	$\text{Max}(1.0, \exp(1.80(\text{水燃比}-0.40)))$
注：蒸汽比为注入蒸汽与进口空气流量的质量比，水燃比为注入水与燃料的质量比	

表 2.4 燃烧系统维修间隔(运行时间)的燃料系数

燃料	A_f
天然气	1.0
轻油	1.5（使用 DLN 时为 2.5）
原油	2.5
渣油	2.5
注：在亚先导预混模式， $A_f=10$	

表 2.5 燃烧系统维修间隔(运行时间)的负荷系数

负荷类型	A_p
基本负荷	1.0
尖峰负荷	$\exp(0.018 \times \text{尖峰负荷燃烧温度增加值})$
注：尖峰负荷燃烧温度增加值的单位为华氏度（F）	

2.1.6 燃烧系统因素启动次数的确定

燃烧系统因素启动次数由公式(2.4)给出

$$CI_{\text{start}} = \frac{CI_{\text{max_start}}}{\frac{\sum_1^N (K_i \times A_{fi} \times A_{ti} \times A_{pi} \times A_{si})}{N}} \quad (2.4)$$

式中：

- i — 启动中的第 i 次
- $CI_{\text{max_start}}$ — 基准情况下的燃烧系统最大维修间隔启动次数
从表 2.1 查出
- K — 水\蒸汽注入系数，根据表 2.6 确定
- A_f — 燃料系数，根据表 2.7 确定
- A_t — 跳闸系数，根据表 2.8 确定
- A_p — 负荷系数，根据表 2.9 确定
- A_s — 启动类型系数，根据表 2.10 确定
- N — 启动次数

表 2.6 燃烧系统维修间隔(启动次数)的水/蒸汽注入系数

运行方式	K
注入蒸汽、干控制曲线	$\text{Max}(1.0, \exp(0.34(\text{蒸汽比}-1.00\%)))$
注入蒸汽、湿控制曲线	$\text{Max}(1.0, \exp(0.34(\text{蒸汽比}-0.50\%)))$
注入水、干控制曲线	$\text{Max}(1.0, \exp(1.80(\text{水燃比}-0.40)))$
注入水、湿控制曲线	$\text{Max}(1.0, \exp(1.80(\text{水燃比}-0.20)))$
注：蒸汽比为注入蒸汽与进口空气流量的质量比，水燃比为注入水与燃料的质量比	

表 2.7 燃烧系统维修间隔(启动次数)的燃料系数

燃料	A_f
天然气	1.0
轻油	1.25（使用 DLN 时为 1.5）
原油	2.0
渣油	3.0

表 2.8 燃烧系统维修间隔(启动次数)的跳闸系数

跳闸	A_f
----	-------

正常停机（无跳闸）	1.0
跳闸	$0.5 + \exp(0.0125 \times \text{跳闸负荷比})$
注：跳闸负荷比为跳闸时的负荷与基本负荷的比值	

表 2.9 燃烧系统维修间隔(启动次数)的负荷系数

负荷类型	A_p
基本负荷	1.0
尖峰负荷	$\exp(0.009 \times \text{尖峰负荷燃烧温度增加值})$
注：尖峰负荷燃烧温度增加值的单位为华氏度（F）	

表 2.10 燃烧系统维修间隔(启动次数)的启动类型系数

启动类型	A_s
正常启动	1.0
快速带负荷启动	1.2
紧急启动	3.0

2.1.7 转子因素运行时间的确定

转子因素运行时间与燃气轮机组的类型有关系

对于 F 级重型燃气轮机组

$$\text{Roter}_{\text{hour}} = \frac{\text{Roter}_{\text{max_hour}}}{\frac{H + 2 \times P}{H + P}} \quad (2.5)$$

对于 E 级重型燃气轮机组

$$\text{Roter}_{\text{hour}} = \frac{\text{Roter}_{\text{max_hour}}}{\frac{H + 2 \times P + 2 \times \text{TG}}{H + P}} \quad (2.6)$$

式中：

$\text{Roter}_{\text{hour}}$ — 基准情况下的转子最大维修间隔运行时间，从表 2.1 查出

H — 基本负荷运行小时数

P — 尖峰负荷运行小时数

TG — 盘车小时数

2.1.8 转子因素启动次数的确定

转子因素启动次数也与燃气轮机组的类型有关系，对于 F 级重型燃气轮机组由公式(2.7)给出

$$\text{Roter}_{\text{start}} = \frac{\text{Roter}_{\text{max_start}}}{\frac{F_h \cdot N_h + F_{w1} \cdot N_{w1} + F_{w2} \cdot N_{w2} + F_c \cdot N_c + F_t \cdot N_t}{N_h + N_{w1} + N_{w2} + N_c}} \quad (2.7)$$

式中：

$\text{Roter}_{\text{max_start}}$ — 基准情况下的转子最大维修间隔启动次数，从表 2.1 查出

N_h — 热启动次数

N_{w1} — 一型暖启动次数

N_{w2} — 二型暖启动次数

N_c — 冷启动次数

N_t — 跳闸次数

F_h — 热启动系数

F_{w1} — 一型暖启动系数

F_{w2} — 二型暖启动系数

F_c — 冷启动系数

F_t — 跳闸系数

以上启动类型和启动系数通过表 2.11 查取。

表 2.11 启动类型和启动系数的关系

启动类型	停机时间	启动系数	
		快速启动	正常启动
热启动 A	0~1 小时	4.0	2.0
热启动 B	1~4 小时	1.0	0.5
一型暖启动	4~20 小时	1.8	0.9
二型暖启动	20~40 小时	2.8	1.4
冷启动	>40 小时	4.0	2.0
跳 闸		4.0	4.0

2.1.9 燃气轮机设备的更换间隔

在 2.1.3 节到 2.1.8 节详细给出了燃烧系统、热通道和转子的由因素运行时间和因素运行小时决定的维修间隔的计算模型和方法。这样我们就可以得到燃烧系统、热通道和转子所涉及的零件的维修时间，从而在保证安全性的情况下得到了最大的可用率，实现了按需维修。通过维修可以解决因结垢、结渣或者其他因素引起的如热效率下降，稳定性下降等影响到生产的现象，使机组的经济性和稳定性增加。

但是也应该认识到，机械零件总是有一定的寿命的，维修虽然可以延长设备的使用寿命，而维修效果却随着维修次数的增加而降低；同时维修也需要一定的维修费用，并且维修费用是随着维修次数的增加而增加的。GE 公司对燃气轮机

第一级喷嘴的维修效果和维修费用统计如图 2.3 所示。

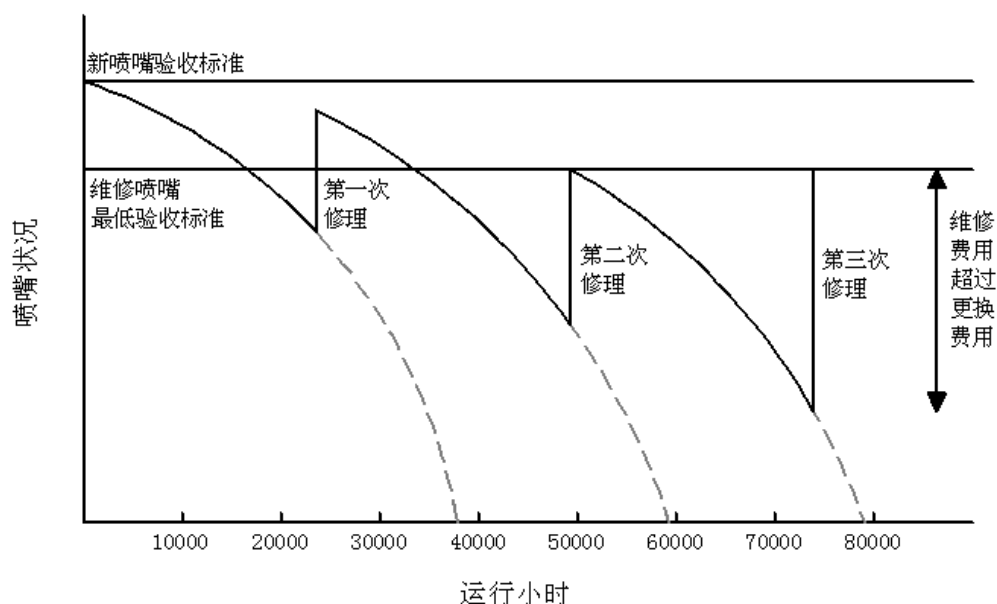


图 2.3 燃气轮机第一级喷嘴的维修效果和维修费用示意图

当维修不能达到设备的预期效果或者维修费用大于更换费用时,应该选择更换设备而不是继续维修设备。设备的更换间隔可以和维修间隔联系起来,这样在安全性得到保证的情况下简化了更换间隔的计算。F 型机组设备的维修和更换间隔的详细关系由表 2.12 给出。需要注意的是这个更换间隔与维修间隔的关系是受到联合循环机组运行状况史和技术修理水平影响的。

表 2.12 GE 9F 型重型燃气轮机部件更换与维修间隔

部件	维修间隔	更换间隔 (运行时间)	更换间隔 (启动次数)
火焰筒	CI	$5 \times CI$	$5 \times CI$
端盖	CI	$5 \times CI$	$5 \times CI$
过度段	CI	$5 \times CI$	$5 \times CI$
燃料喷嘴	CI	$3 \times CI$	$3 \times CI$
联焰管	CI	$1 \times CI$	$1 \times CI$
尾盖		$6 \times CI$	$3 \times CI$
一级喷嘴	HGPI	$2 \times HGPI$	$2 \times HGPI$
二级喷嘴	HGPI	$2 \times HGPI$	$2 \times HGPI$
三级喷嘴	HGPI	$3 \times HGPI$	$3 \times HGPI$
一级复环	HGPI	$2 \times HGPI$	$2 \times HGPI$
二级复环	HGPI	$2 \times HGPI$	$2 \times HGPI$
三级复环	HGPI	$3 \times HGPI$	$3 \times HGPI$

排气室	HGPI		
一级叶片	HGPI	$2 \times \text{HGPI}$	$2 \times \text{HGPI}$
二级叶片	HGPI	$1 \times \text{HGPI}$	$3 \times \text{HGPI}$
三级叶片	HGPI	$3 \times \text{HGPI}$	$3 \times \text{HGPI}$

2.2 在线性能计算的原理和模型

2.2.1 在线性能计算概述

半山发电厂联合循环机组辅助决策系统的研究对象为半山发电厂的 GE 生产的以天然气为燃料的 9F 级联合循环机组，该套机组为我国引进的第一套 9F 级单轴联合循环机组，主体由一台燃气轮机、一台余热锅炉、一台蒸汽轮机和一台发电机组成，见图 2.4，额定装机容量为 389MW。

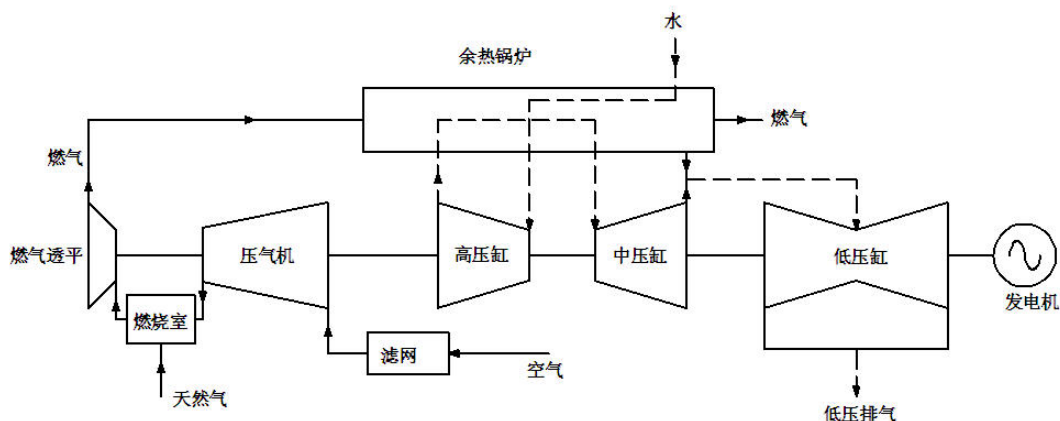


图 2.4 单轴联合循环机组原则性热力系统图

根据联合循环机组的实际情况和在线性能计算的需要，将联合循环机组及其数学模型分为三个部分，即燃气轮机、余热锅炉和蒸汽轮机。燃气轮机又由压气机、燃烧室和燃气透平组成；蒸汽轮机由高压缸、中压缸和低压缸组成。在进行计算的时候将分别对每一个部分进行建模和计算。余热锅炉部分由于设备比较多，而且在现场又实际测点缺乏等因素的制约，所以对余热锅炉只进行整体建模和计算。

我国现有的联合循环机组在线性能计算系统大多数是根据多轴机组进行设计和建模的^{[2][5]}，对于多轴机组，燃气轮机和蒸汽轮机的出力能够单独被测量出来，因此大多数联合循环机组在线性能计算系统单独计算出了两者的气耗、热耗等性能参数，并且使用各自的经验公式对联合循环在非标准状态和变工况下进行修正。而本系统研究的联合循环机组由于燃气轮机和蒸汽轮机共轴，不能实际区分和测量燃气轮机和蒸汽轮机各自的功率，因而在性能计算模型的建立上有一定

不同的特点^{[6][7][8]}。所以本系统将不单独对燃气轮机和蒸汽轮机进行气耗、热耗等性能参数的计算,也不在变工况下单独对它们进行修正。只对整个燃气轮机组的气耗、热耗等性能参数进行计算和进行变工况下的修正。这种计算方法保证了电厂对性能监测及其效果的实际需要,又不需要对电厂控制系统进行改变。

2.2.2 燃气轮机组的模型和计算方法

从图 2.4 中我们可以看出燃气轮机组是联合循环机组的核心组成部分,由压气机、燃烧室和燃气透平组成。空气首先在轴流式压气机中被压缩到一定的压力后送入燃烧室,在燃烧室中与喷射进来的天然气进行混合燃烧,燃烧后产生的燃气进入燃气透平,把其带有的热能部分转换为转子的动能,从而输出机械功,其工作循环如图 2.5 所示。1-2s-3-4s 为燃气轮机组的由四个可逆过程组成的理想循环,1-2-3-4 为考虑到不可逆性后的实际循环。为了求取压气机和燃气透平的相对内效率,必须知道 2s 点和 4s 点的温度,而这两点的温度不能通过实际测点测量,

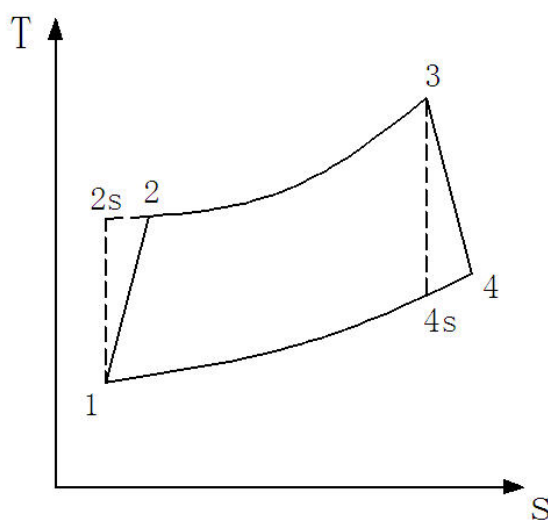


图 2.5 燃气轮机组工作循环 T - s 图

只能根据等熵过程温度与对数相对压力的关系求取。首先根据 t_1 查燃气热力性质表得到 1 点的对数相对压力 $\ln \pi_1$, 根据公式(2.8)

$$\ln \pi_{2s} = \ln \pi_1 + \ln \frac{p_2}{p_1} \quad (2.8)$$

再得到 2s 点对数相对压力以后,再查燃气热力性质表得到 t_{2s} 。4s 点温度的计算方法与此类似。

燃气轮机组压气机、燃烧室和燃气透平的计算既是联系在一起的,但又具有独立性。表 2.13 详细列出了燃气轮机组压气机的性能计算的方法。

表 2.13 压气机性能计算方法

参数	符号	单位	计算方法
大气温度	t_{air}	K	实测
大气压力	p_{air}	kPa	实测
压气机进口温度	t_1	k	实测
压气机进口压力	p_1	kPa	实测

进口滤网压损	$\Delta p_{\text{InletFilter}}$		$\frac{p_{\text{air}} - p_1}{p_{\text{air}}}$
压气机进口焓	h_1	kJ/kMol	根据 t_1 查表
压气机出口温度	t_2	K	实测
压气机出口压力	p_2	kPa	实测
压气机出口焓	h_2	kJ/kMol	根据 t_2 查表
压气机压比	ε_{cp}		p_2/p_1
压气机进口 对数相对压力	$\ln \pi_1$		根据 t_1 查表
压气机理论出口 对数相对压力	$\ln \pi_{2s}$		公式(2.8)
压气机理论出口温度	t_{2s}	K	根据 $\ln \pi_{2s}$ 查表
压气机理论出口焓	h_{2s}	kJ/kMol	根据 t_{2s} 查表
压气机等熵效率	η_{cp}		$\frac{h_{2s} - h_1}{h_2 - h_1}$
空气质量流量	G_{air}	10^3 kg/hr	实测
天然气质量流量	G_{fuel}	10^3 kg/hr	实测
空气摩尔流量	GM_{air}	10^3 kMol/hr	$G_{\text{air}}/M_{\text{air}}$
天然气摩尔流量	GM_{fuel}	10^3 kMol/hr	$G_{\text{fuel}}/M_{\text{fuel}}$
压气机耗功	w_{cp}	kW	$\frac{1000}{3600} \cdot GM_{\text{air}} \cdot (h_2 - h_1)$

燃气轮机计算的难点在于燃气透平入口温度 T_3 的确定, T_3 是整个燃气轮机循环乃至联合循环中的温度最高点,代表了整个联合循环的温度水平。随着 T_3 的升高,燃气轮机组和联合循环的效率会大幅升高,因此提高燃气透平入口温度 T_3 也是燃气轮机和联合循环的发展方向之一^[1]。但是 T_3 的增加主要受到两个因素的制约,第一个是燃气透平特别是前几级喷嘴和叶片材料的温度上限,第二是高燃烧温度下控制氮氧化物生成的燃烧技术^[11]。因此目前的重型燃气轮机设备中,通常会从压气机中抽取部分气体对燃气透平的喷嘴进行冷却,这也导致了 T_3 有着不同的定义。无论哪种定义下的 T_3 温度均比较高,一般的燃气轮机组不直接通过仪器对他进行测量。文献^[12]总结了间接测量的计算 T_3 的一些方法。本系统选择了通过燃烧室的能量平衡关系来计算 T_3 ,这是根据(1)燃烧室是联合循环中最稳定的设备之一,可以根据制造厂提供的数据确定其燃烧效率,一般稳定在98%左右,这也是本系统采用的燃烧室效率;(2)压气机出口空气温度 T_2 存在测点;(3)燃烧室的燃烧效率还可以通过如测量的燃烧产物中的可燃物质的含量等其他方法来计算。燃烧室的能量平衡关系公式为

$$GM_{\text{air}} \cdot h_2 + GM_{\text{fuel}} \cdot Q_l \cdot \eta_{\text{cb}} = (GM_{\text{air}} + GM_{\text{fuel}}) \cdot h_3 \quad (2.9)$$

式中:

GM_{air} — 燃烧室的空气摩尔流量

h_2 — 压气机出口空气焓

GM_{fuel} — 燃烧室的燃料摩尔流量

Q_1 — 燃料的低位发热量

η_{cb} — 燃烧室的燃烧效率

h_3 — 燃气透平进口燃气焓

在确定 GM_{air} 、 h_2 、 GM_{fuel} 、 h_3 、 Q_1 和根据制造厂提供的资料假设 η_{cb} 后, 通过公式(2.9)便可以得到燃气透平进口燃气焓 h_3 , 然后根据燃气的热力性质与温度的关系得到燃气透平入口温度。

表 2.14 详细列出了燃烧室的性能计算的方法。

表 2.14 燃烧室性能计算方法

参数	符号	单位	计算方法
天然气温度	t_{fuel}	K	实测
天然气低位发热量	Q_1	kJ/kMol	实测
燃烧室燃烧效率	η_{cb}		给定
燃气透平进口焓	h_3	kJ/kMol	公式(2.9)
燃气透平进口温度	t_3	K	根据 h_3 查表
燃气透平进口压力	p_3	kPa	实测
燃烧室压损	Δp_{cb}		$\frac{p_3 - p_2}{p_3}$
理论燃料空气比	Δp_{cb}		$\frac{h_3 - h_{\text{fuel}}}{Q_1 + h_2 - h_3}$

通过燃烧室能量平衡关系计算得到的 T_3 实际上是燃烧室出口的燃气温度, 采用这种方法定义的 T_3 不包括冷却空气对它的影响。加上等熵效率的定义和通常的计算方法中均不考虑抽气对它的影响, 因此模型在压气机和燃烧室计算等熵效率的过程中忽略冷却空气因素, 而是认为压气机进口空气均通过燃烧室燃烧后进入燃气透平。

根据前面的分析我们得到可以根据燃气透平进口温度 t_3 和等熵过程中温度与对数相对压力的关系来求取燃气透平理论出口温度 t_{4s} , 从而计算燃气透平的相对内效率, 表 2.15 详细列出了燃气透平的性能计算的方法。

表 2.15 燃烧透平性能计算方法

参数	符号	单位	计算方法
燃气透平出口温度	t_4	K	实测
燃气透平出口压力	p_4	kPa	实测
燃气透平压比	ε_{gt}		p_3/p_4
燃气透平出口焓	h_4	kJ/kMol	根据 t_4 查表
燃气透平进口 对数相对压力	$\ln \pi_3$		根据 t_3 查表
燃气透平理论出口 对数相对压力	$\ln \pi_{4s}$		$\ln \pi_{4s} = \ln \pi_3 - \ln \frac{p_3}{p_4}$

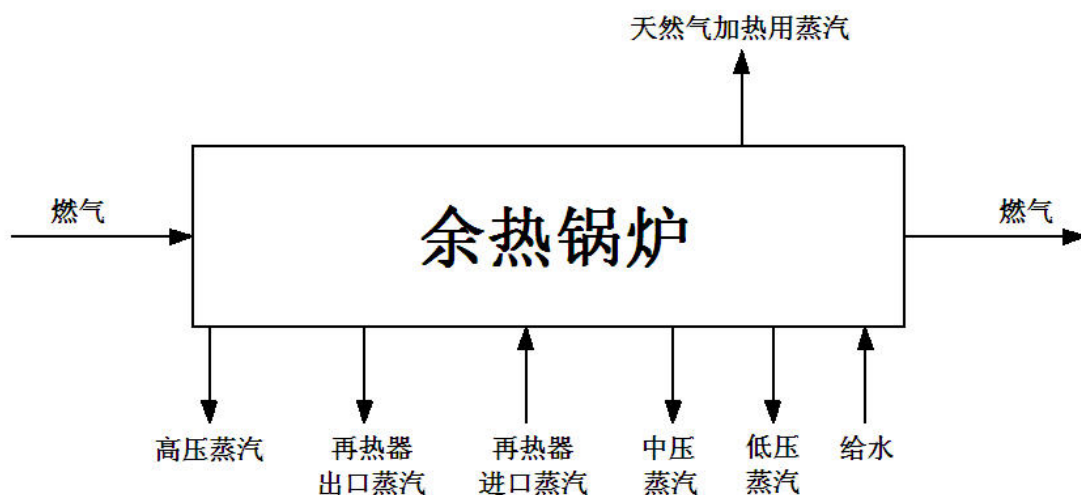
燃气透平理论出口温度	t_{4s}	K	根据 $\ln \pi_{4s}$ 查表
燃气透平理论出口焓	h_{4s}	kJ/kMol	根据 t_{4s} 查表
燃气透平相对内效率	η_{gt}		$\frac{h_3 - h_4}{h_3 - h_{4s}}$
燃气透平功率	w_{gt}	kW	$\frac{1000}{3600} \cdot (GM_{air} + GM_{fuel}) \cdot (h_3 - h_4)$

得到压气机、燃烧室和燃气透平的参数和性能以后,可以计算燃气轮机组整体的功率、效率性能参数。在 2.2.1 节说明了因为该系统面向的对象是单轴联合循环机组,燃气轮机和蒸汽轮机各自的功率不能直接测量,但是这里通过燃气透平的出力和压气机的耗功推算出了燃气轮机组的功率。

表 2.16 燃气轮机组性能计算方法

参数	符号	单位	计算方法
燃气轮机组功率	w_{gtUnit}	kW	$w_{gt} - w_{cp}$
燃气轮热效率	η_{gtUnit}		$3.6 \cdot \frac{w_{gtUnit}}{GM_{fuel} \cdot Q_l}$
燃气轮机组热耗率	q_{gtUnit}	kJ/kwh	$1000 \cdot \frac{GM_{fuel} \cdot Q_l}{w_{gtUnit}}$
燃气轮机组气耗率	d_{gtUnit}	m^3/kwh	$q_{gtUnit} / Q_l \times 22.4$

2.2.3 余热锅炉的模型和计算方法



联合循环机组的余热锅炉一般有不补燃式和补燃式两种,根据研究,不补燃式相比补燃式具备投资费用低、运行可靠性高和启动快等特点^[1]。半山发电厂联合循环机组也采用了不补燃式的余热锅炉,即外界不向余热锅炉供给除燃气轮机排气之外的能量,余热锅炉通过高温高压排气带来的热量,加热蒸汽循环中的热水。

半山联合循环机组由于出力达到 390MW，同时采用三压不补燃方式，因而余热锅炉的结构比较复杂，包括了高压、中压和低压的各种换热设备，根据电厂人员对在线性能监测的实际需要和实际测点缺乏等因素的制约，本系统将不详细对每个换热设备进行建模和计算，而是把余热锅炉看成一个整体进行建模和计算。

半山发电厂联合循环机组采用的是三压余热锅炉，余热锅炉同时输出高压蒸汽、中压蒸汽和低压蒸汽，高压蒸汽和低压蒸汽进入汽轮机做功，而中压蒸汽与蒸汽轮机高压缸出口的蒸汽汇合后进入余热锅炉再热器再热，因此没有在图 2.4 的联合循环机组原则性热力系统图标出中压蒸汽。

根据余热锅炉简化模型图得到余热锅炉中的能量平衡关系，

$$G_{\text{gas}}h_{\text{gasIn}} = G_{\text{bhp}}(h_{\text{bhp}} - h_{\text{fw}}) + G_{\text{brh}}(h_{\text{brhOut}} - h_{\text{brhIn}}) + G_{\text{bip}}(h_{\text{bip}} - h_{\text{fw}}) + G_{\text{blp}}(h_{\text{blp}} - h_{\text{fw}}) + G_{\text{fhw}}(h_{\text{fhw}} - h_{\text{fw}}) + G_{\text{gas}}h_{\text{gasOut}} \quad (2.10)$$

式中：

- G_{gas} — 余热锅炉烟气流量
- h_{gasIn} — 余热锅炉进口烟气焓
- h_{gasOut} — 余热锅炉出口烟气焓
- G_{bhp} — 高压蒸汽流量
- h_{bhp} — 高压蒸汽焓
- h_{fw} — 给水焓
- G_{brh} — 再热蒸汽流量
- h_{brhIn} — 再热器进口蒸汽焓
- h_{brhOut} — 再热器出口蒸汽焓
- G_{bip} — 中压蒸汽流量
- h_{bip} — 中压蒸汽焓
- G_{blp} — 低压蒸汽流量
- h_{blp} — 低压蒸汽焓
- G_{fhw} — 天然气加热蒸汽流量
- h_{fhw} — 天然气加热蒸汽焓

通过能量平衡关系，我们可以得到锅炉的正效率和反效率。表 2.17 详细列出了余热锅炉的性能计算的方法。

表 2.17 余热锅炉性能计算方法

参数	符号	单位	计算方法
余热锅炉进口 烟气温度	t_{gasIn}	K	实测
余热锅炉进口烟气焓	h_{gasIn}	kJ/kMol	根据 t_{gasIn} 查表
余热锅炉出口 烟气温度	t_{gasOut}	K	实测

余热锅炉出口烟气焓	h_{gasOut}	kJ/kMol	根据 t_{gasout} 查表
余热锅炉烟气流量	G_{gas}	10^3 kMol/hr	实测
高压蒸汽流量	G_{bhp}	10^3 kMol/hr	实测
高压蒸汽温度	t_{bhp}	K	实测
高压蒸汽压力	p_{bhp}	kPa	实测
高压蒸汽焓	h_{bhp}	kJ/kg	根据 t_{bhp} 和 p_{bhp} 查表
再热蒸汽流量	G_{brh}	10^3 kMol/hr	实测
再热器进口蒸汽温度	t_{brhIn}	K	实测
再热器进口蒸汽压力	p_{brhIn}	kPa	实测
再热器进口蒸汽焓	h_{brhIn}	kJ/kg	根据 t_{brhIn} 和 p_{brhIn} 查表
再热器出口蒸汽温度	t_{brhOut}	K	实测
再热器出口蒸汽压力	p_{brhOut}	kPa	实测
再热器出口蒸汽焓	h_{brhOut}	kJ/kg	根据 t_{brhOut} 和 $p_{\text{brhOutbhp}}$ 查表
中压蒸汽流量	G_{bip}	10^3 kMol/hr	实测
中压蒸汽温度	t_{bip}	K	实测
中压蒸汽压力	p_{bip}	kPa	实测
中压蒸汽焓	h_{bip}	kJ/kg	根据 t_{bip} 和 p_{bip} 查表
低压蒸汽流量	G_{blp}	10^3 kMol/hr	实测
低压蒸汽温度	t_{blp}	K	实测
低压蒸汽压力	p_{blp}	kPa	实测
低压蒸汽焓	h_{blp}	kJ/kg	根据 t_{blp} 和 p_{blp} 查表
天然气加热蒸汽流量	G_{fhw}	10^3 kMol/hr	实测
天然气加热蒸汽温度	t_{fhw}	K	实测
天然气加热蒸汽压力	p_{fhw}	kPa	实测
天然气加热蒸汽焓	h_{fhw}	kJ/kg	根据 t_{fhw} 和 p_{fhw} 查表
给水温度	t_{fw}	K	实测
给水压力	p_{fw}	kPa	实测
给水蒸汽焓	h_{fw}	kJ/kg	根据 t_{fw} 和 p_{fw} 查表
锅炉的吸热量	Q_s	kJ/kg	$G_{\text{bhp}}(h_{\text{bhp}} - h_{\text{fw}}) + G_{\text{brh}}(h_{\text{brhOut}} - h_{\text{brhIn}}) + G_{\text{bip}}(h_{\text{bip}} - h_{\text{fw}}) + G_{\text{blp}}(h_{\text{blp}} - h_{\text{fw}}) + G_{\text{fhw}}(h_{\text{fhw}} - h_{\text{fw}})$
余热锅炉的正效率	$\eta_{\text{boilerPos}}$		$\frac{Q_s \times M_{\text{gas}}}{h_{\text{gasIn}}}$
余热锅炉的反效率	$\eta_{\text{boilerRev}}$		$1 - \frac{h_{\text{gasOut}}}{h_{\text{gasIn}}}$

2.2.4 蒸汽轮机的模型和计算方法

燃气—蒸汽联合循环中,提高整机效率的一个重要方法就是提高燃气轮机组排气中余热的利用程度,所以要尽可能地降低余热锅炉的排气温度,如果从蒸汽

轮机中抽气加热给水无疑导致了余热锅炉排气温度的上升,降低了联合循环机组效率。同时,目前以天然气为燃料的联合循环机组,余热锅炉的排气温度在 80 ~ 90℃,与凝汽器中的凝结水温差不多,因此联合循环中使用的蒸汽轮机与常规电站中使用的蒸汽轮机有所不同,它们一般没有回热系统和给水加热器^[1],即热力系统不从蒸汽轮机中抽取蒸汽来加热给水。

如上所述原因,半山发电厂联合循环机组也采用了无回热的蒸汽轮机,因此蒸汽轮机的高压缸、中压缸和低压缸中各自的蒸汽流量保持不变,这简化了每个缸的相对内效率的计算过程。由于连接余热锅炉各个加热器和蒸汽轮机各个汽缸间的管道中蒸汽的流量和焓没有变化或变化很小,本系统采用高压蒸汽流量和焓作为高压缸蒸汽流量和进口蒸汽焓,再热器蒸汽流量做为中压缸蒸汽流量和进口蒸汽焓,再热蒸汽和低压蒸汽的流量和作为低压缸的蒸汽流量。

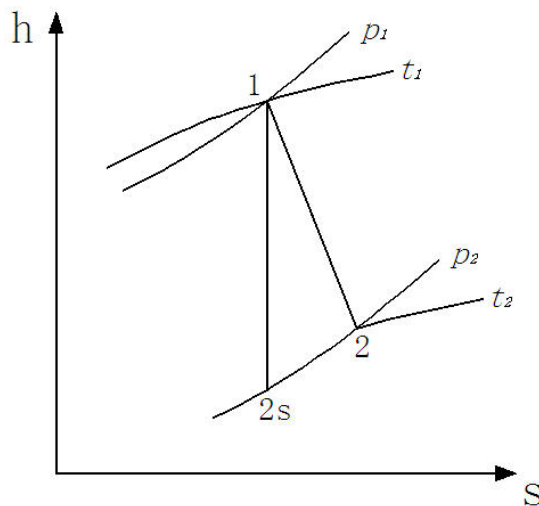


图 2.7 蒸汽在汽轮机缸内热力过程

蒸汽在汽轮机每个缸的热力过程线为从图 2.7 所示的 1 点膨胀做功到 2 点, 2_s点为假设的等熵膨胀过程终点。根据水和水蒸汽的热力性质可以知道, 对于任意一点, 只要知道它的两个热力参数, 就能得到其他的参数, 因此根据 1 点的温度和压力计算得到 1 点的熵。又因为 2_s点的熵与 1 相同, 再根据 p₂ 查取 2_s点的熵。得到点 1、2 和 2_s的熵之后就能计算每个缸的相对内效率。

$$\eta = \frac{h_1 - h_2}{h_1 - h_{2s}} \quad (2.11)$$

式中:

- η — 汽轮机高(中、低)压缸效率
- h_1 — 汽轮机高(中、低)压缸进口蒸汽焓
- h_2 — 汽轮机高(中、低)压缸出口蒸汽焓
- h_{2s} — 汽轮机高(中、低)压缸理论出口蒸汽焓

与燃气轮机的性能计算一样,通过热力计算得到不能直接测量的蒸汽轮机功率。2.18 详细列出了蒸汽轮机的性能计算的方法。

表 2.18 蒸汽轮机性能计算方法

参数	符号	单位	计算方法
高压缸出口蒸汽温度	$t_{sthPOut}$	K	实测

高压缸出口蒸汽压力	$p_{sthpOut}$	kPa	实测
高压缸出口蒸汽焓	$h_{sthpOut}$	kJ/kg	根据 $t_{sthpOut}$ 和 $p_{sthpOut}$ 查表
高压缸相对内效率	η_{hp}		公式(2.11)
中压缸出口蒸汽温度	$t_{stipOut}$	K	实测
中压缸出口蒸汽压力	$p_{stipOut}$	kPa	实测
中压缸出口蒸汽焓	$h_{stipOut}$	kJ/kg	根据 $t_{stipOut}$ 和 $p_{stipOut}$ 查表
中压缸相对内效率	η_{ip}		公式(2.11)
低压缸进口蒸汽温度	t_{stlpIn}	K	实测
低压缸进口蒸汽压力	p_{stlpIn}	kPa	实测
低压缸进口蒸汽焓	h_{stlpIn}	kJ/kg	根据 t_{stlpIn} 和 p_{stlpIn} 查表
低压缸出口蒸汽温度	$t_{stlpOut}$	K	实测
低压缸出口蒸汽压力	$p_{stlpOut}$	kPa	实测
低压缸出口蒸汽焓	$h_{stlpOut}$	kJ/kg	根据 t_{stlpIn} 和 p_{stlpIn} 查表
低压缸相对内效率	η_{lp}		公式(2.11)
蒸汽轮机的功率	w_{st}	kw	$[G_{bhp} \cdot (h_{bhp} - h_{sthpOut}) + G_{brh} \cdot (h_{brhOut} - h_{stipOut}) + (G_{brh} + G_{blp}) \cdot (h_{stlpIn} - h_{stipOut})] \times \frac{1000}{3600}$

2.2.5 联合循环机组的性能计算与修正

通过上面的计算得到了联合循环机组燃气轮机、余热锅炉和蒸汽轮机三大部件的主要性能参数,通过这些性能参数,用户可以分别判断每个设备的运行情况。但是为了从整体上把握联合循环机组的运行情况,就需要计算联合循环机组的热效率、热耗率和气耗率,这些指标是衡量整个机组能量利用程度和经济性的重要指标。本系统气耗率的单位是 m^3/kwh ,与通常使用的 kg/kwh 不同是因为半山的天然气使用体积计费,因而采用体积单位。

表 2.19 联合循环机组性能计算方法

参数	符号	单位	计算方法
联合循环计算功率	w_{cc}^1	kw	$w_{gtUnit} + w_{st}$
联合循环实际功率	w_{cc}	kw	实测
联合循环功率相对误差	Δw_{cc}		$\frac{w_{cc}^1 - w_{cc}}{w_{cc}^1}$
联合循环机组热效率	η_{cc}		$3.6 \times \frac{w_{cc}}{Q_l \cdot GM_{fuel}}$
联合循环机组热耗率	q_{cc}	kJ/kwh	$1000 \cdot \frac{GM_{fuel} \cdot Q_l}{w_{cc}}$
联合循环机组气耗率	d_{cc}	m^3/kwh	$q_{cc}/Q_l \times 22.4$

联合循环机组的出力受大气环境的影响很大,根据 GE 提供的资料^[13],在冬季(大气温度 4℃,大气压力 102.2kPa,大气湿度 75%)机组的基准负荷能够达

到 412MW，而到了夏季（大气温度 28.5℃，大气压力 100.0kPa，大气湿度 79%）机组的基准符合只能达到 355MW，两者绝对差值达到 57MW，相对差值达到 13.8%。因此应该把实时计算得到的联合循环机组的关键参数进行大气温度、压力和湿度的修正，从而可以综合比较不同时期联合循环的机组性能和运行水平，提高机组的经济性。

除了大气环境的影响之外，发电机频率、天然气低位发热量和总运行小时数都对联合循环机组的功率有影响，所以要分别对每个因素进行修正。

由于本系统的研究对象为单轴联合循环机组，因此对于燃气轮机和蒸汽轮机单独进行性能修正意义不大，而且具有较大难度，虽然可以通过热力计算得到燃气轮机和蒸汽轮机的功率，但是根据计算的结果分析，与实际情况比较还存在一定的误差。从整体上对联合循环机组的关键性能进行修正，充分满足了电厂对于核心性能和参数的需要。

评价一台联合循环机组性能的关键参数是它的功率、热耗率和气耗率，因为热耗率和气耗率成正比关系，因此本系统只进行大气环境等因素对联合循环的功率和热耗率修正，再根据修正的热耗率计算出对应的修正气耗率。修正计算过程中使用到的经验公式基于制造厂提供的变工况曲线^[13]拟合得到。下面式(2.12)~(2.22)列出了各因素对功率影响的拟合公式；式(2.13)~(2.28)列出了各因素热耗率影响的拟合公式。某些参数在拟合公式中的单位与上述性能计算方法中的单位不一致，因此在进行修正计算时要注意进行单位换算。

大气温度 t (℃) 对联合循环机组功率的影响 F_t^w

$$F_t^w = \begin{cases} -6.31407 \cdot 10^{-7}t^3 - 1.16993 \cdot 10^{-5}t^2 - 2.32229 \cdot 10^{-3}t + 1.05145 & t < 15^\circ\text{C} \\ -2.03858 \cdot 10^{-7}t^3 - 1.4429 \cdot 10^{-5}t^2 - 4.32074 \cdot 10^{-3}t + 1.08061 & t \geq 15^\circ\text{C} \end{cases} \quad (2.12)$$

大气压力 p (kPa) 对联合循环机组功率的影响 F_p^w

$$F_p^w = 5.70293 \cdot 10^{-8}x^3 - 2.17661 \cdot 10^{-5}x^2 + 1.24566 \cdot 10^{-2}x - 9.58128 \cdot 10^{-2} \quad (2.13)$$

大气相对湿度 Φ (%) 对联合循环机组功率的影响 F_Φ^w ， t 表示大气温度

$$F_{\Phi}^w = \begin{cases} -1.07689 \cdot 10^{-9} \Phi^3 - 2.58699 \cdot 10^{-8} \Phi^2 + 3.92558 \cdot 10^{-5} \Phi + 9.97628 \cdot 10^{-1} & t = 15^{\circ}\text{C} \\ -7.52794 \cdot 10^{-11} \Phi^3 - 1.71822 \cdot 10^{-9} \Phi^2 + 1.31078 \cdot 10^{-5} \Phi + 9.99004 \cdot 10^{-01} & t = 16^{\circ}\text{C} \\ -1.93733 \cdot 10^{-11} \Phi^3 - 4.1729 \cdot 10^{-08} \Phi^2 + 3.65646 \cdot 10^{-6} \Phi + 9.99983 \cdot 10^{-1} & t = 25^{\circ}\text{C} \\ -6.55443 \cdot 10^{-10} \Phi^3 + 3.21756 \cdot 10^{-8} \Phi^2 - 1.69032 \cdot 10^{-5} \Phi + 1.00147 & t = 30^{\circ}\text{C} \\ 1.35586 \cdot 10^{-9} \Phi^3 - 4.36242 \cdot 10^{-7} \Phi^2 - 1.27463 \cdot 10^{-5} \Phi + 1.00307 & t = 35^{\circ}\text{C} \\ 1.47043 \cdot 10^{-9} \Phi^3 - 4.14262 \cdot 10^{-7} \Phi^2 - 6.3074 \cdot 10^{-5} \Phi + 1.00689 & t = 40^{\circ}\text{C} \end{cases} \quad (2.14)$$

发电机频率 f (Hz) 对联合循环机组功率的影响 F_f^w , t 表示大气温度

$$F_f^w = \begin{cases} 9.44962515 \cdot 10^{-7} f^4 + 1.11546815 \cdot 10^{-3} f^3 - 1.87201571 \cdot 10^{-1} f^2 + 9.88672045 f - 1.7067163 \cdot 10^2 & t = 10^{\circ}\text{C} \\ -4.49324812 \cdot 10^{-4} f^4 + 9.02120695 \cdot 10^{-2} f^3 - 6.7969479 f^2 + 2.27796013 \cdot 10^2 f - 2.86465949 \cdot 10^3 & t = 25^{\circ}\text{C} \\ -1.559916833 \cdot 10^{-3} f^4 + 3.170534533 \cdot 10^{-1} f^3 - 2.416889148 \cdot 10 f^2 + 8.189940683 \cdot 10^2 f - 1.040867616 \cdot 10^4 & t = 40^{\circ}\text{C} \end{cases} \quad (2.15)$$

天然气低位发热量 Q_l (kJ/kg) 对联合循环机组功率的影响 F_Q^w , x 表示氢碳元素原子数比

$$F_Q^w = \begin{cases} 1.91152 \cdot 10^{-11} Q^2 - 2.29199 \cdot 10^{-6} Q + 1.06297 & x = 3.57 \\ 1.14147 \cdot 10^{-11} Q^2 - 1.55866 \cdot 10^{-6} Q + 1.04883 & x = 3.95 \\ 9.30997 \cdot 10^{-12} Q^2 - 1.36034 \cdot 10^{-6} Q + 1.04455 & x = 4 \end{cases} \quad (2.16)$$

排汽压力 p_l (kPa) 对联合循环机组功率的影响 $F_{p_l}^w$

$$tF_{p_l}^w = 1.7578837 p_l^5 - 6.10601646 \cdot 10 p_l^4 + 8.4847885 \cdot 10^2 p_l^3 - 5.98532996 \cdot 10^3 p_l^2 + 1.96275465 \cdot 10^4 p_l - 2.20435856 \cdot 10^4 \quad (2.17)$$

$$F_{p_l}^w = tF_{p_l}^w(p_l) - tF_{p_l}^w(5.96) \quad (2.18)$$

天然气加热器进口天然气温度 t_g ($^{\circ}\text{C}$) 对联合循环机组功率的影响 $F_{t_g}^w$

$$F_{t_g}^w = -4.30891 \cdot 10^{-5} t_g^3 + 2.64268 \cdot 10^{-3} t_g^2 + 1.82188 t_g - 2.77773 \cdot 10 \quad (2.19)$$

发电机功率因数 P_f 对联合循环机组功率的影响 $F_{P_f}^w$, w_{cc} 表示联合循环机组功率

$$tF_{P_f}^W = \begin{cases} -2.0005 \cdot 10^{-10} w_{cc}^5 + 2.538 \cdot 10^{-7} w_{cc}^4 - 1.08768 \cdot 10^{-4} w_{cc}^3 + \\ \quad 3.1215 \cdot 10^{-2} w_{cc}^2 - 1.76212 \cdot 10^{-2} w_{cc} + 1.475 \cdot 10^3 & P_f = 0.85 \\ -2.47421 \cdot 10^{-10} w_{cc}^5 + 2.23671 \cdot 10^{-7} w_{cc}^4 - 6.48474 \cdot 10^{-5} w_{cc}^3 + \\ \quad 1.79586 \cdot 10^{-2} w_{cc}^2 + 5.23689 \cdot 10^{-1} w_{cc} + 1.47500 \cdot 10^3 & P_f = 0.90 \\ 3.00737 \cdot 10^{-10} w_{cc}^5 - 3.15320 \cdot 10^{-7} w_{cc}^4 + 1.14175 \cdot 10^{-4} w_{cc}^3 - \\ \quad 6.04102 \cdot 10^{-3} w_{cc}^2 + 1.09760 w_{cc} + 1.47500 \cdot 10^3 & P_f = 0.95 \\ -3.78647 \cdot 10^{-10} w_{cc}^5 + 4.90730 \cdot 10^{-7} w_{cc}^4 - 2.2161 \cdot 10^{-4} w_{cc}^3 + \\ \quad 4.86615 \cdot 10^{-2} w_{cc}^2 - 2.25292 w_{cc} + 1.475 \cdot 10^3 & P_f = 1.0 \end{cases} \quad (2.20)$$

$$F_{P_f}^W = tF_{P_f}^W(0.85) - tF_{P_f}^W(P_f) \quad (2.21)$$

机组点火总小时 H 对联合循环机组功率的影响 F_H^W

$$F_t^W = \begin{cases} 0.8907 \left(\frac{H}{1000} \right)^3 - 2.28 \left(\frac{H}{1000} \right)^2 + 2.4285 \left(\frac{H}{1000} \right) + 0.0036 & H \leq 1200 \\ 0.030209 \times 0.999988^H \times x^{0.513797} & H > 1200 \end{cases} \quad (2.22)$$

大气温度 t ($^{\circ}\text{C}$) 对联合循环机组热耗率的影响 F_t^q

$$F_t^q = \begin{cases} -3.53169 \cdot 10^{-7} t^4 + 4.76307 \cdot 10^{-6} t^3 + 2.98383 \cdot 10^{-5} t^2 & t < 15^{\circ}\text{C} \\ \quad -3.58425 \cdot 10^{-3} t + 1.06061 & \\ 4.03643 \cdot 10^{-7} t^3 + -4.73603 \cdot 10^{-5} t^2 + -3.66284 \cdot 10^{-3} t^2 & t \geq 15^{\circ}\text{C} \\ \quad +1.07595 & \end{cases} \quad (2.23)$$

大气压力 p (kPa) 对联合循环机组热耗率的影响 F_p^q

$$F_p^q = -7.82219 \cdot 10^{-7} p^2 + 1.00548 \cdot 10^{-2} p - 8.54334 \cdot 10^{-3} \quad (2.24)$$

大气相对湿度 Φ (%) 对联合循环机组热耗率的影响 F_{Φ}^q , t 表示大气温度

$$F_{\Phi}^q = \begin{cases} -1.0461 \cdot 10^{-9} \Phi^3 + 1.60486 \cdot 10^{-8} \Phi^2 + 4.37927 \cdot 10^{-5} \Phi & t = 15^{\circ}\text{C} \\ \quad +9.96979 \cdot 10^{-1} & \\ -2.01977 \cdot 10^{-13} \Phi^3 - 1.11025 \cdot 10^{-8} \Phi^2 + 2.62487 \cdot 10^{-5} \Phi & t = 16^{\circ}\text{C} \\ \quad 9.97988 \cdot 10^{-01} & \\ 1.35571 \cdot 10^{-12} \Phi^3 - 3.50454 \cdot 10 \Phi^2 + 2.79673 \cdot 10^{-5} \Phi & t = 25^{\circ}\text{C} \\ \quad +9.98002 \cdot 10^{-1} & \\ -5.88142 \cdot 10^{-10} \Phi^3 + 2.81346 \cdot 10^{-8} \Phi^2 + 1.9795 \cdot 10^{-5} \Phi & t = 30^{\circ}\text{C} \\ \quad +9.98551 \cdot 10^{-1} & \\ 7.29726 \cdot 10^{-10} \Phi^3 - 2.81812 \cdot 10^{-7} \Phi^2 + 3.19479 \cdot 10^{-5} \Phi & t = 35^{\circ}\text{C} \\ \quad +9.98873 \cdot 10^{-1} & \\ 1.31082 \cdot 10^{-9} \Phi^3 - 3.51307 \cdot 10^{-7} \Phi^2 + 8.93462 \cdot 10^{-6} \Phi & t = 40^{\circ}\text{C} \\ \quad +1.00086 & \end{cases} \quad (2.25)$$

发电机频率 f (Hz) 对联合循环机组功率的影响 F_f^q , t 表示大气温度

$$F_f^q = \begin{cases} 1.7715196151 \cdot 10^{-5}f^4 - 2.4879749415 \cdot 10^{-3}f^3 + 1.02694686 \cdot 10^{-1}f^2 & t = 10^\circ\text{C} \\ -4.6031931397 \cdot 10^{-1}f - 3.2443859056 \cdot 10 & \\ -3.90189833 \cdot 10^{-4}f^4 + 7.83325664 \cdot 10^{-2}f^3 - 5.90120134f^2 & t = 25^\circ\text{C} \\ 1.97747319 \cdot 10^2f - 2.48624693 \cdot 10^3 & \\ -1.181755029 \cdot 10^{-3}f^4 + 2.407218273 \cdot 10^{-1}f^3 - 1.839100776 \cdot 10f^2 & t = 40^\circ\text{C} \\ +6.24605868 \cdot 10^2f - 7.95603347 \cdot 10^3 & \end{cases} \quad (2.26)$$

天然气低位发热量 Q (kJ/kg) 对联合循环机组功率的影响 F_Q^q , x 表示氢碳元素原子数比

$$F_Q^q = \begin{cases} 7.50041 \cdot 10^{-12}Q^2 - 1.07731 \cdot 10^{-6}Q + 1.03213 & x = 3.57 \\ 7.21482 \cdot 10^{-12}Q^2 - 1.05111 \cdot 10^{-6}Q + 1.03407 & x = 3.95 \\ 7.15356 \cdot 10^{-12}Q^2 - 1.04531 \cdot 10^{-6}Q + 1.03423 & x = 4 \end{cases} \quad (2.27)$$

机组点火总小时 H 对联合循环机组功率的影响 F_H^q

$$F_H^q = \begin{cases} 0.392 \cdot 10^{-3}H + 0.0053 & H \leq 1200 \\ \frac{-0.00187 \times 1598.57 + 2.325613H^{0.852128}}{1598.57 + H^{0.852128}} & H > 1200 \end{cases} \quad (2.28)$$

通过拟合的公式(2.12)~(2.22)得到了大气参数、发电机频率、天然气低位发热量、排气压力、天然气加热器进口天然气温度、发电机功率因数和总点火小时对联合循环机组功率的修正因素。通过公式(2.23)~(2.28)得到了大气参数、发电机频率、天然气低位发热量和总点火小时对联合循环机组热耗率的修正因素。从而可以将不同时间不同参数运行时的功率和热耗率转化为 ISO 工况的功率、热耗率和气耗率, 比较机组的运行水平和进行经济性分析。式(2.29)是把功率(kw)修正到 ISO 工况的公式, 式(2.30)是把热耗率(kJ/kwh)修正到 ISO 工况的公式, 式(2.31)根据修正热耗率计算修正气耗率 (m^3/kwh)。

$$w_{cc}^{ISO} = (w_{cc} - F_{P_f}^w - F_{t_g}^w - F_{p_l}^w) \cdot \frac{1}{F_t^w \cdot F_p^w \cdot F_\phi^w \cdot F_f^w \cdot F_Q^w} \cdot \left(1 + \frac{F_H^w}{100}\right) \quad (2.29)$$

$$q_{cc}^{ISO} = \frac{1000 \cdot GM_{fuel} \cdot Q_l \cdot F_t^q \cdot F_p^q \cdot F_\phi^q \cdot F_f^q \cdot F_Q^q}{w_{cc}^{ISO} \cdot F_H^q} \quad (2.30)$$

$$d_{cc}^{ISO} = \eta_{cc}^{ISO} / Q_l \times 22.4 \quad (2.31)$$

第三章 程序设计框架与技术在系统中的应用

3.1 系统的开发平台—Microsoft .NET

因为半山发电厂的三台服务器均使用 Microsoft 的 Windows 2003 Enterprise Edition 操作系统，所以辅助决策系统的开发平台使用了同样来自 Microsoft 的 .NET 框架，保证开发出来的辅助决策系统与操作系统达到最好的兼容性和最佳的性能。

.NET 框架（.NET Framework）的第一个版本即 .NET Framework 1.0 由微软公司于 2002 正式推出，经过 3 年多的发展，微软于 2005 年 11 月推出了 .NET Framework 2.0，相比 .NET 框架 1.0，无论是通用语言运行时还是其他组成部分都作出了许多重要的改变。伴随着其最新操作系统 Windows Vista 在 2006 年 11 月的发布，微软同时推出了 .NET Framework 3.0，但是与其他两个版本有所不同的是 .NET 框架

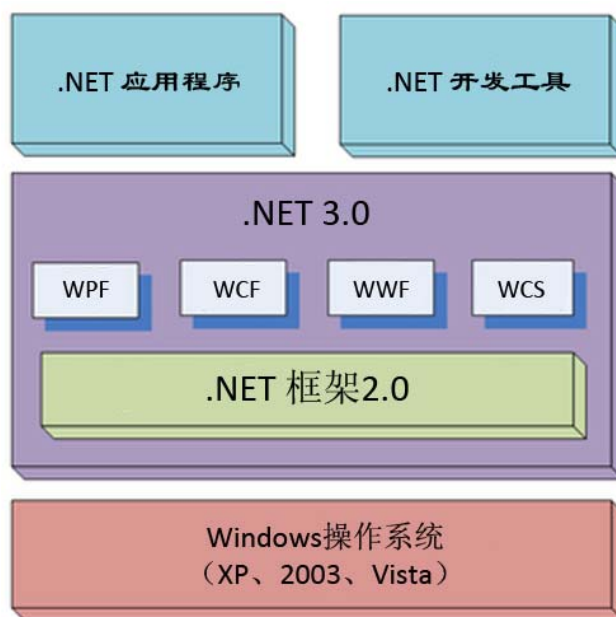


图 3.1 .NET 框架 3.0 架构

3.0 在其核心方面并没有变化，它仍然使用 .NET Framework 2.0，只是在其之上引入了四种基本的新技术，分别为 Windows 描述基础（Windows Presentation Foundation）、Windows 通信基础（Windows Communication Foundation）、Windows 工作流基础（Windows Workflow Foundation）和 Windows CardSpace，并将这四个技术作为其他应用程序的开发和运行的基础。如图 3.1 所示

由于 .NET 框架 3.0 发布时间较短，并且它仍然使用 2.0 版本的核心，所以在半山发电厂联合循环机组辅助决策系统中将采用 .NET Framework 2.0 作为开发平台。

3.1.1 .NET 框架概述

.NET 是一种革命性的开发技术和开发模式，它与 Win32 API 或者 COM 一样，是把系统服务以接口形式提供给开发人员的软件开发平台。与以往不同的是，.NET 框架能够更好地完成代码重用、资源配置、多语言集成、开发和安全

管理等任务,在安全性、易用性以及开发效率等方面远远超过了以前的开发模式。这个平台从设计之初,就对与一些长期困扰软件设计和开发的老问题从根本上进行了重新考虑,并且给出了非常好的解决方案。比如引入强命名组件集来避免了 COM 中的 DLL 版本和更新问题,从而简化了应用程序的部署和升级。

.NET 框架核心由两个部分构成:公共语言运行时(Common Language Runtime, CLR)和 Framework 类库(Framework Class Library, FCL)^{[14][15]}。公共语言运行时提供了所有类型的应用程序都要使用的编程模型,包括它自己的文件加载器、内存管理器(垃圾收集器)、安全系统(代码访问安全性)和线程池等。除此之外,CLR 还提供了一个面向对象的编程模型,它定义了类型和对象是什么,以及它们的行为方式。Framework 类库则提供了所有应用程序模型都要使用的一个面向对象的 API 集合。利用其中包含的类型定义,开发者可以执行文件和网络 I/O、调度其他线程上的任务、画图、比较字符串等。

正是基于公共语言运行时和 Framework 类库的核心,.NET 框架提供了一个一致的编程模式,它为所有类型的应用程序提升了抽象等级。这就意味着无论软件开发人员是想构建控制台应用程序、图形应用程序、网站、还是构建供其他应用程序使用的组件,只需学习和掌握一个编程模型和一套 API。而不是像以前为了编写一个网站,必须去学习 Javascript 和 vbscript,而为了编写桌面应用程序,又必须去学习 MFC 或者 Delphi。因此半山发电厂联合循环机组辅助决策系统只使用了.NET 框架便完成了用于后台计算的服务器端桌面应用程序和用于人机交互的网站 B/S 系统。

同样是因为基于公共语言运行时和 Framework 类库,.NET 框架实现了不同编程语言的集成,在.NET 中不同的语言可以集成在一起编写同一个解决方案、同一个工程甚至同一个类。与 COM 要求的具有二进制兼容特性的语言才能协作不同,.NET 的语言集成实现只需要语言符合.NET 公共语言规范(Common Language Specification, CLS),并且使用通用类型系统(Common Type System, CTS)。这样就可以充分发挥每种语言的特点,Microsoft 自身已经提供了几个编译器来生成面向 CLR 的代码:C++/CLI、C#和 Visual Basic .NET 等。在半山发电厂联合循环机组辅助决策系统中,需要.NET 框架进行 Office (OWC) 开发,由于 OWC 是由 COM 技术构建的,其中的很多函数和方法使用了缺省参数,而.NET 框架中的 Visual Basic .NET 正好支持这一个特性而 C#不支持,因此这一部分使用 Visual Basic .NET 来编写,从而提高了开发效率。

.NET 框架提供了一个面向对象的 API 集合,但是由于 Windows 及数以万计的运行在其上面的软件发展了很长的时间,这个集合并不可能包罗万象的,它也在不断的发展过程中,因此某些组件可能并不在这个集合中,特别是很多第三方软件只提供了 COM 接口甚至于函数。这种情况下.NET 框架提供了 interop 功能,也叫 P/Invoke,支持访问现有的 COM 组件和调用现有的 DLL 中的 Win32 或者

其他函数。如半山发电厂联合循环机组辅助决策系统访问 PI 实时数据库时,.NET 框架中并没有提供直接的 API, 而是必须通过 P/Invoke 调用 PI-API。

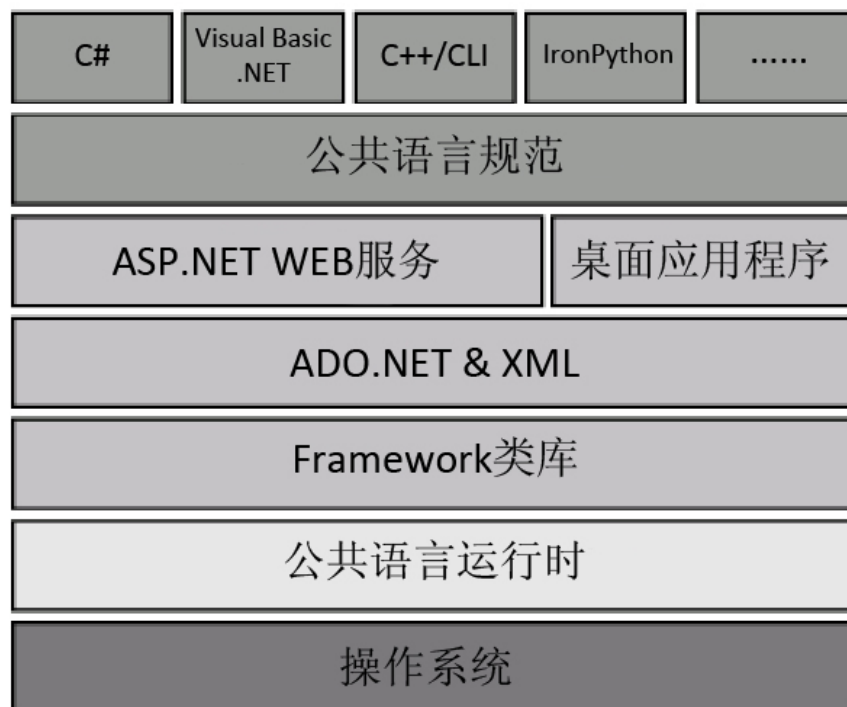


图 3.2 .NET 框架架构

以上特性可以很直观的从 .NET 框架的架构图 3.2 中得到。

微软免费提供了 .NET 框架、.NET SDK 以及面向 CLR 的 C#、Visual Basic .NET 和 C++/CLI 等语言的编译器,从而可以使用任何文本编辑器来创建应用程序,但是这样开发效率将很低。而使用集成开发环境 (IDE) 则可以高效率的进行项目开发,支持 .NET 开发的 IDE 有微软的 Visual Studio、开源的 SharpDeveloper、MonoDeveloper 等。半山发电厂联合循环机组辅助决策系统的选择 Visual Studio 2005 作为开发软件和环境。

3.1.2 C#程序设计语言

.NET 框架具有与语言无关的特性,可以使用任何符合语言符合 .NET 公共语言规范,并且使用通用类型系统的语言。但是由于 C#语言是专门为 .NET 框架而设计,充分利用了 CLR 和 CLI 提供的特性,因此在 .NET 程序设计中,C#是被使用最多的语言,半山发电厂联合循环机组辅助决策系统也将使用 C#语言作为基本语言,只是在一部份 OWC 开发时使用 Visual Basic .NET。

C#语言随着 .NET 框架 2.0 版本的发布而升级,在 .NET 框架 3.0 中 C#的版本仍然为 2.0。它是一种设计简单、现代的、面向对象和面向组件的程序设计语言,语法风格与 C/C++相似,语义上大量吸收了其他语言如 DELPHI 和 JAVA 的特性。

因为基于公共语言运行时，C#并不直接被编译器编译为计算机可直接执行的二进制代码，而是与其他.NET 语言一样被编译成中间语言(Intermediate Language)。程序执行时中间语言再被公共语言运行时编译为计算机可执行的二进制代码。所以经过编译的 C#程序扩展名虽然是.exe 可执行文件，但是在没有安装.NET 框架的计算机上是不能执行的。

作为 C 系列的语言，C#相比 C/C++在许多方面进行了增强和限制，从而使其简化了内存管理、面向对象和面向组件。下面是 C#语言的一些特点。

C#不需要对象类型进行显示释放，而能够自动进行垃圾回收，避免了内存泄露的问题。对象访问一般通过安全的引用实现，指针只能在特别声明的不安全语句中被使用，并且编译时要打开不安全选项。

C#不支持多重继承，只允许单一继承，这样就避免了程序员因为对多重继承使用不当而造成的复杂性。但是 C#允许类继承多个接口。

C#是一个完全的面向对象的程序设计语言，所有类均直接或者简介的继承自 System.Object 类，因而在 C#程序中不存在全局变量以及像 C 语言中的 printf() 这样的全局函数。

C# 2.0 中引入了泛型的概念来代替 C++中的模板，与 C++模板不同，C#泛型参数是在运行阶段而不是编译阶段初始化的，因此能够实现跨语言。C#泛型还支持一些 C++模版不支持的特性如通过接口对泛型参数进行类型限制，但是它又限制了表达式不能像在 C++模版中一样被用于泛型参数。

C# 2.0 中还支持局部类型，即可以将一个类分在不同的文件中，从而可以使用多种语言来编写同一个类。

3.1.3 WEB 开发与 ASP.NET

半山发电厂联合循环机组辅助决策系统中使用 B/S 系统来负责处理人机交互的界面，这就要求进行 WEB 开发，在前面提到了.NET 框架为程序设计提供了一致的编程模式，其中与 WEB 网站开发相关的就是 ASP.NET。

ASP.NET 是一个统一的 Web 开发模型，可用来开发网站、WEB 应用程序和 WEB 服务。它是微软.NET 框架的一部份，因此应用 ASP.NET 进行开发时，可以访问.NET 框架中的类，也可以使用与公共语言运行库兼容的任何语言来编写代码。

传统的 WEB 开发技术 ASP 和 PHP 是基于脚本的，如 ASP 采用 vbscript 或者 Javascript，脚本语言一般通过解释方式运行，语句在运行时刻由脚本解释器一条条解释执行。而 ASP.NET 基于.NET 框架用面向 CLR 的语言编写，因此他们都是被编译成中间语言，由公共语言运行时再把他们编译成二进制代码，公共语言运行时编译中间语言时，会将编译过的代码缓存起来，在第二次使用时，

CLR 不会重新编译代码，而是直接运行已编译过的缓存代码，这样就很大的提高了运行效率；同时 ASP.NET 的编译运行方式让开发人员在编译时检查代码是否有语法错误、类型安全问题以及其他问题，减少了运行时刻程序的测试时间，增加了开发效率。此外脚本语言一般对面向对象的支持比较弱，而半山发电厂联合循环机组辅助决策系统中 ASP.NET 开发使用的 C#语言则是完全面向对象的，这样就可以在 B/S 系统开发中使用多态、继承和其他等面向对象的特性来简化开发过程。

在 ASP.NET 框架中，一个网页通常分为两个部分：可视元素和逻辑。可视元素由一个包含静态标记的文件组成，这些静态标记包括 HTML 元素和 ASP.NET 服务器控件，网页的逻辑就是程序代码，它们规定了这个网页的规则。可视元素和逻辑可以放在一个文件里面，但是通常把它们分开在不同的文件中，可视元素所在文件扩展名为 `aspx`，代码文件的扩展名即所使用语言的文件扩展名，如使用 C#语言，那么扩展名就是 `cs`。这样就实现页面美工与后台程序的分离，方便网站的开发和维护。由于 .NET 框架提供一致的编程模式，与进行 Windows 桌面程序开发相似。ASP.NET 同样使用事件驱动的编程模型，如按钮能够响应点击事件，列表框能够响应选项改变事件等。当用户的浏览器向服务器发出页面请求时，服务器端控件被 ASP.NET 引擎处理成 HTML，生成的 HTML 和原来页面中的 HTML 元素被发送到用户的浏览器中。当用户点击按钮时，将会触发一个点击事件，ASP.NET 引擎会捕捉到这个事件并执行相应的事件处理代码，并将更新后的页面重新发送到浏览器。ASP.NET 页面的事件响应如图 3.3。

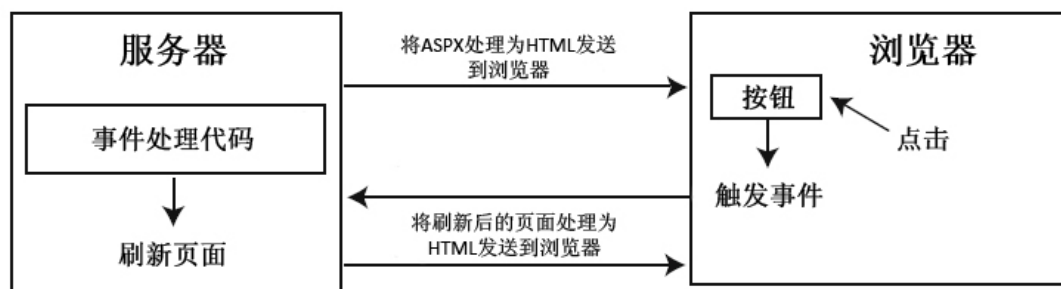


图 3.3 ASP.NET 页面的事件响应

几乎每个应用程序都需要进行用户和权限管理，因此在 ASP.NET 2.0 中内置了成员/角色管理 (Membership/Role Management)^[17]，两者相互联系又是相互独立的，成员管理提供了一种验证和存储用户凭据的方法，利用它可以实现网站用户的创建和修改、管理密码和对访问站点的用户进行身份验证等。角色管理则提供了管理授权的方法，通过它可以指定应用程序中的特定用户访问特定的资源。把角色管理和用户管理结合起来则可以向角色分配用户，从而将同一个角色的用户组视为一个单元，为他们分配资源和权限。在半山发电厂联合循环机组辅助决

策系统中存在系统管理员、维修间隔计算管理员和在线性能计算管理员等不同的角色，它们的职责是不一样的，因此在这里系统通过 ASP.NET 2.0 中内置了成员/角色管理来分配权限。用户管理和角色管理各自又是相互独立的，修改用户的资料不会改变它的角色，同时修改用户的角色不会改变它的密码等资料。

ASP.NET 2.0 除开内置了角色/成员管理，还提供了一组与成员管理相关的服务器端控件，因此只需要在开发过程中进行简单的配置和拖动这些控件便可以实现网站的用户管理。但是由于提供的这些服务器端控件又一些局限性并且不支持角色管理，因此在半山发电厂联合循环机组辅助决策系统中不采用这种方式，而是利用角色管理和权限管理提供的 API 结合 WEB 服务来进行系统的用户和权限管理。

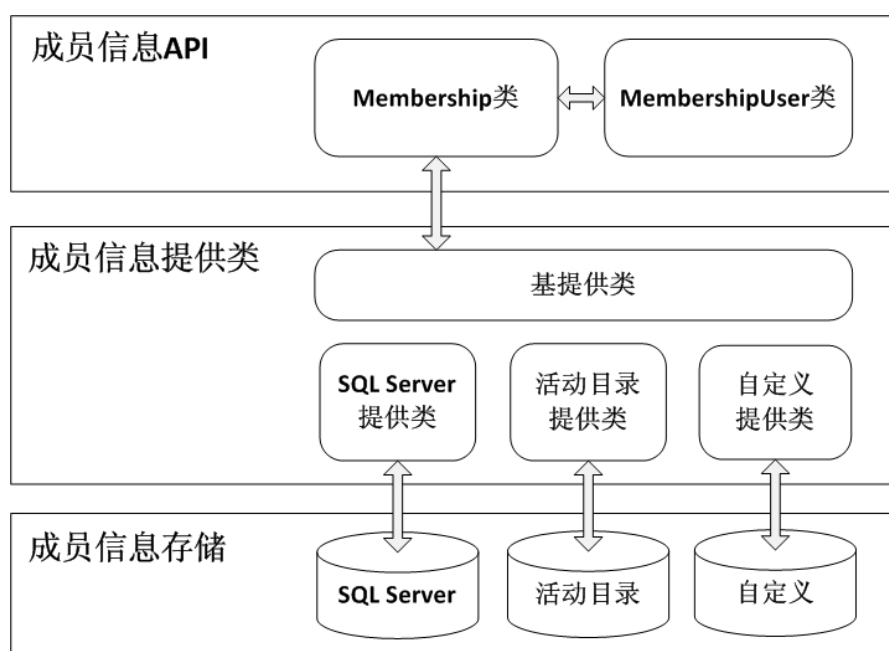


图 3.4 ASP.NET 成员管理的架构

ASP.NET 成员管理由成员管理 API、提供类和信息数据三者组成，其架构如图 3.4 所示。位于最下层的是信息数据的存储方法如数据库和活动目录等，中间为数据信息提供类（provider），这里使用了一个工厂设计模式，不同的数据信息提供类均继承自基提供类，从而保证了最上层的成员管理 API 与最底层数据信息的无关性，即无论你使用哪一种方法存储数据，都是使用同样的类的同样方法来进行成员管理。最上层的成员管理 API 由 Membership 类和 MembershipUser 类组成，实现用户的添加、获取和认证。

ASP.NET 角色管理架构与成员管理类似，最上层同样提供了角色管理 API 用于添加修改角色和分配用户到角色，角色管理 API 主要由 Roles 类组成。

结合半山电厂的实际情况和安全性的需要，半山发电厂联合循环机组辅助决策系统选择使用 SQL Server 数据库来存放成员/角色信息。这就要求在使用之前

配置数据库,.NET 框架 SDK 提供了 aspnet_regsql.exe 来自动生成成员/角色数据库所需要的表、视图和存储过程。

为了弥补成员/角色管理的不足,半山发电厂联合循环机组辅助决策系统定义了一个 `MemberShipUserWrapper` 类,该类继承自 `MembershipUser`,这样他就拥有了 `MembershipUser` 类保护和公有属性与方法,如成员用户名、密码和 Email 等。此外还给 `MemberShipUserWrapper` 类添加了新的属性和方法、例如 `Role` 属性,通过 `MemberShipUserWrapper.Role` 就可以直接得到用户所属的角色,而这个实现在系统默认的 API 中是没有的。

ASP.NET 不仅可以用来开发网站和 WEB 应用程序,还可以开发 WEB 服务。WEB 服务是一种可以用来解决跨网络应用集成问题的开发模式,它在网络上公布可编程接口,其他应用程序或者另外一个 WEB 服务可以通过这个接口来完成该 WEB 服务所提供的功能。由于 WEB 服务的协议如 HTTP、XML 和 SOAP 等都是标准化的,所以可以使用任何语言和任意平台来编写 WEB 服务,也可以在任何地方无论是 WEB 应用程序还是桌面应用程序甚至移动设备来调用 WEB 服务。一定程度上 WEB 服务可以看成 B/S 模式的发展,因为它相比 C/S 模式和 B/S 模式,他进一步降低了系统的耦合程度。

在半山发电厂联合循环机组辅助决策系统中运用了 WEB 服务结合成员管理 API 来进行网站用户的身份验证,网站通过调用 WEB 服务提供的身份验证接口来进行用户登录判断,这样可以方便以后系统的扩展,如在桌面程序或者其他地方也一样可以通过调用该 WEB 服务进行身份验证。半山发电厂联合循环机组辅助决策系统网站没有采用传统的 WEB 服务调用方法,而是通过 Ajax 来调用提供的 WEB 服务。

身份验证登录的 WEB 服务代码 (C#):

```
using System;
using System.Web;
using System.Collections;
using System.Web.Services;
using System.Web.Services.Protocols;
using System.Web.Script.Services;
using System.Web.Security;

namespace CCADMS.Web.Services
{
    [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
    [ScriptService]
    public class AuthenticationService : System.Web.Services.WebService
    {
        [WebMethod]
        public bool bsLogin(string name,string passord,bool createPersistentCookie)
        {
```

```

        bool isValidate = Membership.ValidateUser(name, passord);
        if(isValidate==true)
        {
            SetAuthCookie(name, createPersistentCookie);
        }
        return isValidate;
    }

    [WebMethod]
    public void Logout()
    {
        FormsAuthentication.SignOut();
    }

    public void SetAuthCookie(string userName, bool createPersistentCookie)
    {
        FormsAuthentication.SetAuthCookie(userName, createPersistentCookie);
    }
}

```

网站调用 WEB 服务的部分代码（Javascript）：

```

function ValidateUser()
{
    CCADMS.Web.Services.AuthenticationService.IsValidationCode(ValidationCode.value,Is
    ValidationCodeCompleted);
}

//用于校核验证码
function IsValidationCodeCompleted(result)
{
    if(result==true){
        CCADMS.Web.Services.AuthenticationService.bsLogin(username.value,password.value,
        chkRemember.checked,ValidateUserCompleted);
    }
    else{
        ValidationCode.select();
        document.selection.clear();
        ValidationCode.focus();
        feedBack.innerHTML = "验证码错误";
    }
}

function ValidateUserCompleted(result)
{
    password.value = "";
    if (result == true)
    {

```

```
        feedBack.innerHTML = "登录成功";  
        window.location.href="default.aspx"  
    }  
    else  
    {  
        feedBack.innerHTML = "登录失败";  
    }  
}
```

3.1.4 数据库与 ADO.NET

半山发电厂联合循环机组辅助决策系统需要使用到普通数据库存储用户信息、测点资料和部分计算结果。系统使用微软的 SQL Server 2000 作为普通数据库。.NET 框架中的组件 ADO.NET 提供了数据库应用程序开发的支持,通过它可以读取和写入数据库中数据。

ADO.NET 是 .NET 框架中数据访问编程模型,是实现应用程序数据读取、数据处理和数据存储的基础,包含了 System.Data 命名空间下的一系列类。ADO.NET 不是它之前常用数据库操作技术 ADO(ActiveX Data Objects)的升级,而是一个全新的完全基于 .NET 框架的数据访问方法。

ADO.NET 可以实现两大功能,第一个功能是数据的获取,第二个功能数据的表示。用于实现数据获得类有 DbConnection、DbCommand、DbDataReader 和 DbDataAdapter。用于数据表示的类有 DataTable、DataRelation 和 DataSet。

DbConnection 类用于实现数据库的连接,指定它的数据库连接字符串以后便可以连接或者断开指定数据库。DbConnection 类是一个抽象类,从它派生了对应具体数据库的数据库连接类,如连接 SQL Server 2000 及后续版本的 SqlConnection 类、连接 Oracle 的 OracleConnection 类和连接 OleDb 数据库的 OleDbConnection 类等,其中 OleDb 不是一个数据库,而是指一种数据库连接方式,通过它你也可以连接 SQL Server 或者 Oracle 数据库,但是通过 OleDbConnection 类连接这些数据库会比使用专门为他们设计的连接类效率低。

DbCommand 类实现与数据库的交互,通过它可以执行 SQL 语句或者运行存储过程并得到返回的结果,它也是一个抽象类,必须指定具体数据库命令类,某些特定的类存在针对相应数据库的方法。

DbDataReader 类可以通过它的 Read 方法读取 DbCommand 类返回的结果,它是顺序读取返回的结果集的,因此它具有很高的效率,通常用它读取结果到一个泛型列表或者对象集合。不同数据库必须扩展 DbDataReader 类来实现特有的操作。

DbDataAdapter 类可以说是 ADO.NET 相比数据库连接框架创新的体现,同

样用于操作数据。可以通过它的 Fill 和 Update 方法非常简单的实现 DataSet 对象和数据库的数据交换从而实现对数据的读取、更新和删除。但是它的执行效率相比 DbDataReader 类低。同样它派生了针对不同数据库的 DataAdapter 类。

DataTable 类是用来表示数据的，他与实现数据库连接和操作的类不同，它不需要派生特别的类来对应不同的数据库。DataTable 类对应于数据库中的表。当在 DataTable 中操作数据是，原始数据表的数据并没有改变，直到你使用它的 AcceptChanges 方法才会更新数据库中的表。

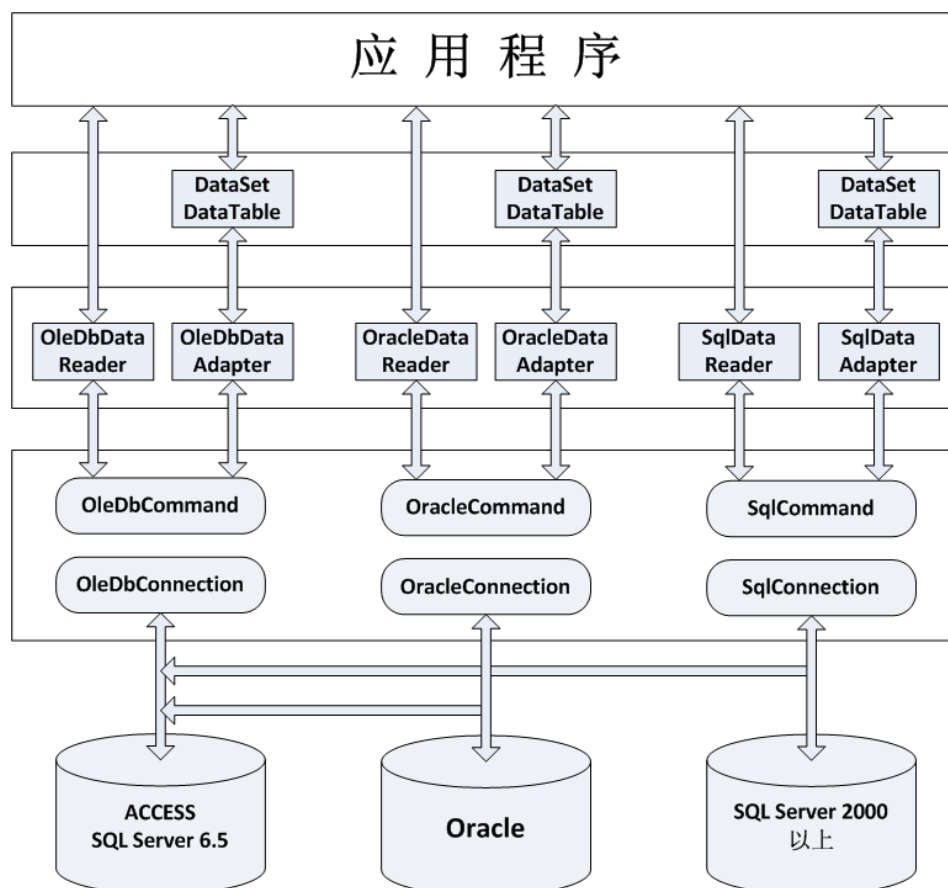


图 3.5 ADO.NET 的架构原理图

DataRelation 类表示数据库中表与表之间的关系和约束。

DataSet 类可以说是 ADO.NET 框架中最重要的类，它可以看作 DataTable 和 DataRelation 对象的集合，相当于一个本地的数据库。其更新原理与 DataTable 类相似。而且 DataSet 可以和 XML 进行转换。

通过上述几个类，ADO.NET 有两套方式实现了 .NET 框架对数据库的操作，如图 3.5 所示一种是利用 DbDataReader 方式实现，另外一种通过 DbDataAdapter、DataSet 方式实现，两者相比，第一种效率比较高，但是读取数据时要保持数据库连接，后一种效率稍低，但是可以使用 ADO.NET 的离线数据操作，即只在需要时连接数据库，并且编程相对简单。

由于在半山发电厂联合循环机组辅助决策系统中使用对象关系映射（ORM）的设计思想和技术（3.3 节），因此系统开发中对数据库的操作使用了 ADO.NET 中连接方式，使用 `DbDataReader` 高效率的读取结果到自定义的由关系数据库映射的对象集合。

在半山发电厂联合循环机组辅助决策系统还使用了微软设计并且开源的 `Data Access Application Block`，它包装并且优化了数据访问代码，让用户可以用更简单的方式访问数据库、执行 SQL 语句或者存储过程并返回 `DbDataReader` 或者 `DataSet`。

以读取半山数据库中用于记录 PI 数据库的测点名的 `CCUnitTag` 表为例，如果不使用 `Data Access Application Block` 的代码（C#）为：

```
string sql = "select * from CCUnitTag where UnitNumber=@UnitNumber";
SqlConnection conn = new SqlConnection();
conn.ConnectionString =
    ConfigurationManager.ConnectionStrings["SQLSERVER"].ConnectionString;
SqlCommand cmd = new SqlCommand();
cmd.Connection = conn;
SqlParameter param = new SqlParameter("@UnitNumber", SqlDbType.Int,4);
param.Value = 1;
cmd.CommandText = sql;
cmd.Parameters.Add(param);
conn.Open();
SqlDataReader rdr = cmd.ExecuteReader();
//对 rdr 进行操作
conn.Close();
```

使用 `Data Access Application Block` 的代码为：

```
string sql = "select * from CCUnitTag where UnitNumber=@UnitNumber";
SqlParameter param = new SqlParameter("@UnitNumber", SqlDbType.Int, 4);
param.Value = 1;
using (SqlDataReader rdr = SqlHelper.ExecuteReader(SqlHelper.ConnectionString,
    CommandType.Text, sql, param)){
    //对 rdr 进行操作
}
```

从这两段的代码比较可以看出，使用 `Data Access Application Block` 可以减少程序中重复的数据库访问代码提高代码复用，并且提高程序的可读性和可维护性。

3.2 PI 实时/历史数据库

3.2.1 SIS 系统与 PI 数据库

为了保证发电机组的安全和经济运行，在机组中通常加装了许多测点来监测电厂的运行状态，对于半山发电厂的每台联合循环机组大约有 8000 个以上的测

点, 这些测点的测量数据送到 DCS 以后, 工作人员便可以通过控制系统得到这些运行参数。这样虽然实现了运行参数的监测, 但是同样具有很大的局限性。首先, 只有 DCS 即控制系统内的设备才能获得数据, 而这个区域是很小而且相对封闭的。其次, DCS 不保存这些运行数据, 而缺乏实际运行历史数据, 就不能再现电厂的运行过程, 从而不能对运行状况进行分析和总结。

为了解决上述局限, 我国科研人员提出了 SIS 系统的概念^[18], SIS 系统能够把数据从 DCS 中取出来, 这样能在更大的范围内让更多工作人员观测到数据, 并把数据存储到数据库中, 工作人员可以根据数据进行分析并且根据分析结果在具备权限的条件下把操作指令发送回到 DCS 去实际控制机组的运行。SIS 系统的核心是存储数据的数据库。由于发电机组数据测点多, 并且为了能够真实监测或者再现运行情况, 取数据间隔比较小, 因此 SIS 中的数据流量非常大, 这就要求在 SIS 系统中使用实时/历史数据库, 这种数据库与常见的关系数据库不同, 关系数据库着重于描述数据之间的关系, 而实时/历史数据库则着重于数据的实时性, 它们的实现原理和算法也有很大的不同。

目前国内使用的实时数据库主要产品有 Wonderware 公司的 Industrial SQL、Honeywell 公司的 PHD 和 OSI 公司的 PI, 其中应用的最广泛的是 PI 实时/历史数据库系统, 半山发电厂的 SIS 系统也是基于 PI。

PI (Plant Information System) 是由美国 OSI Software 公司开发的基于 C/S 架构的工业实时数据集成和应用平台, 可以通过它的客户端 ActiveView 或者提供的编程接口实现 B/S 架构。它是底层控制网络 and 上层管理信息系统网络连接的桥梁。PI 根据时间序列的进行采集数据即数据只在同一时间点存储一次, 同时数据信息以数据原型存储, 保持原有的时间间隔和精度。它的数据处理采用了旋转门压缩技术和二次过滤技术, OSI 公司公布了这两种算法但是对它持有专利。正式基于这样一种技术, 它能够有效的对数据进行压缩从而能够在有限的空间内保存大量的数据。更重要的是这两种技术实现了数据的快速回放, 让用户能够快速访问压缩过的历史数据。

PI 数据库的核心是其中存储的数据, 其它的一切工作都是建立在数据的基础上, 因此必须保证数据的准确性和有效性。要保证数据的准确性除开传感器、接口机等设备要定期检查维护之外, PI 数据库中标签点的属性的设置也非常重要。因为通过标签点的属性可以设置点是否压缩、点的最大压缩时间以及点的量程等。如果设置不当就会造成数据丢失而产生错误。

PI 数据库提供了丰富的客户端工具用于通过各种方式显示和分析数据库中的数据, 主要的客户端工具有 PI DataLink、PI ProcessBook、PI ActiveView 和 RtAlerts 等。除开客户端工具 PI 数据库还提供了编程接口, 用户可以利用 PI-API、PI-SDK 和 PI-OLEDB 等在应用程序中直接读取 PI 数据库中的数据。

半山发电厂联合循环机组辅助决策系统的开发主要使用了 PI 数据库提供的

编程接口 PI-API 和 PI-SDK 以及它的客户端工具 PI-ProcessBook 和 PI-ActiveView。

3.2.2 PI-ProcessBook 与 PI-ActiveView

PI-ProcessBook 是用来组态和显示 PI 数据库中实时/历史数据的主要图形用户接口, PI-ProcessBook 包括两种模式^[19], 第一种是设计模式, 在这种模式下可以利用内置的标签、流程图或者实时和历史趋势图创建并保存与 PI 数据库交互的图形画面。第二种是显示模式, 在这种模式下 PI-ProcessBook 根据用户创建画面显示 PI 系统的实时数据和历史数据, 并且各种画面都可以进行移动和缩放。此外, ProcessBook 中内嵌了 Visual Basic 的子集 Visual Basic for Application^[20], 通过 VBA 可以编写画面或者趋势自动化的脚本程序来扩展 PI-ProcessBook 的应用, 同时集成了 VBA 就意味着可以读取普通关系数据库中数据在 PI-ProcessBook 中显示。

半山发电厂联合循环机组辅助决策系统中主要使用 PI-ProcessBook 来显示在线性能计算的数据, 通过它的趋势图可以很直观的判断机组的运行情况。系统首先在 PI-ProcessBook 的设计模式下利用它提供的组件对机组设备机型组态, 并把部分关键测点的数据和性能计算结果通过标签控件放在相应位置。控件的使用很简单, 只需要指定它的 Tag 点或者自定义数据集即可。自定义数据集是建立在 PI 数据库的基础上, 它允许用户对数据进行加工和处理, 如最常用的单位变换。而且数据集的数据可以直接和显示组件绑定, 好像数据直接来源于数据库。

系统充分利用了 PI-ProcessBook 支持 VBA 开发的功能, 对 PI-ProcessBook 进行扩展。例如, 用户点击显示数据的标签时, 趋势图中便增加用户单击的标签对应的数据或者自定义数据的趋势, 方便用户对数据进行趋势和走向分析。其实现代码 (VBA) 如下

```
Private Sub Display_Click(ByVal IvarX As Long, ByVal IvarY As Long)
    Set smbsSelected = ThisDisplay.SelectedSymbols
    '保证用户只点击了一个控件
    If smbsSelected.Count = 0 Or smbsSelected.Count > 1 Then
        Exit Sub
    End If
    如果点击'Value, 修改趋势图
    For Each smbValue In smbsSelected
        If smbValue.Type = PBObjLib.pbSYMBOLTYPE.pbSymbolValue Then
            Dim tagName As String
            Dim pureTagName As String
            tagName = smbValue.GetTagName(1)
            pureTagName = Right(tagName, Len(tagName) - InStr(3, tagName, "\"))
            tndDetail.TrendTitle = tagName
            If Not (tndDetail.PtCount = 0) Then
                If Not (tndDetail.GetTagName(1) = tagName) Then
```

```
        tndDetail.RemoveTrace (1)
        tndDetail.AddTrace (tagName)
    End If
Else
    tndDetail.AddTrace (tagName)
End If
End If
Next smbValue
End Sub
```

PI 实时/历史数据库系统是一个完全的 C/S 系统，而 PI-ProcessBook 就是这个系统中最常用的客户端工具，是一个标准的 Windows 应用程序。PI 为了在 C/S 系统的基础上支持 WEB 开发，让用户通过浏览器也能够实现对数据库中的数据进行检测，于是发布了 PI-ActiveView 工具。

PI-ActiveView 同样是一个用于显示 PI 数据库中的实时/历史数据的工具，它的特点是能够部署在 Internet 或公司网络中，从而用户可以通过浏览器对它进行浏览。同时它还能集成在电子表格制作工具或者报表工具中。

实质上，PI-ActiveView 是利用 ActiveX 技术把 PI-ProcessBook 制作的显示页面（PDI 文件）嵌入到浏览器或者其他支持 ActiveX 技术的程序中去，用户便可以通过浏览器或者其他程序来查看图形显示页面，从而实现对 PI 数据库中的数据进行监测并进行交互而不需要安装 PI-ProcessBook^[22]，满足了需要浏览访问 ProcessBook 画面但不需要创建画面的用户。

ActiveX 是由微软设计的用于实现对象连接与嵌入的规范，它通过定义容器和组件之间的接口规范实现下述效果：一个遵循规范的控件，可以很方便地在多种容器中使用而不用修改控件的代码；同样，通过实现标准接口调用，在一个遵循规范的容器中可以很容易地嵌入任何遵循规范的控件。在 PI 数据库系统中，PDI 文件中的控件就是满足这样规范的控件，而 PI-ProcessBook 和微软的 Internet Explorer 浏览器就是满足这种规范的容器，正是这样实现了在浏览器中显示 PDI 文件。

PI-ActiveView 利用 ActiveX 实现 PI 数据库 WEB 开发的同时，也受到 ActiveX 技术缺点的限制。第一是不是所有的浏览器都支持或者完全支持 ActiveX 技术，目前只有微软的 Internet Explorer 完全支持 ActiveX 技术，也就是说利用 PI-ActiveView 开发的网站只能通过 Internet Explorer 浏览器浏览，不过现在的 Windows 操作系统都自带了这个浏览器，同时它能够免费下载并且使用于 Mac 操作系统。第二也是最关键的是使用 ActiveX 技术可能存在严重的安全隐患，由于可以在 Internet Explorer 浏览器这种符合规范的容器中使用遵循 ActiveX 接口的控件，也就是说几乎可以执行任何代码而不受限制，这其中就可能包括恶意代码。ActiveX 控件可以通过权威的数字签名来保证其不被篡改，但是实现这种数字签名需要一笔费用。正是因为这个原因 Internet Explorer 浏览器在默认状态下

是不使用未签名的 ActiveX 控件的。考虑到半山发电厂联合循环机组辅助决策系统是在局域网中使用并且为了节省费用，因此其 ActiveX 控件没有使用数字签名，从而在使用 PI-ActiveView 时需要调低 Internet Explorer 浏览器的安全等级。

PI-ActiveView 实现服务器和浏览器的数据传输可以分为 4 个步骤，如图 3.6。

第一步，当浏览器通过网络请求包含 PDI 对象的网页时，网页内容被下载到浏览器所在的计算机。

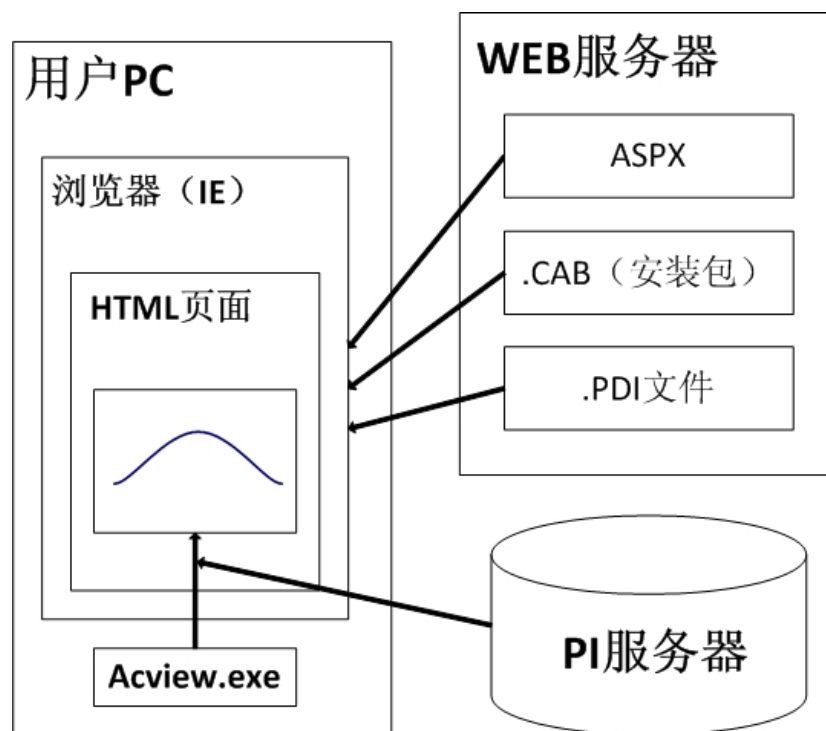


图 3.6 PI-ActiveView 数据显示过程

第二步，浏览器解释网页的时候发现嵌入对象（PDI 文件），首先寻找对象的 Codebase 属性，对应 PDI 文件它指向 PI-ActiveView 组件的安装包。如果 Codebase 属性不存在，浏览器会尝试利用本地计算机中的组件创建一个对象的实例，如果没有 PI-ActiveView 组件，浏览器便不能显示 PDI 文件。如果存在 Codebase 属性，浏览器会检查本地计算机中是否安装 PI-ActiveView 组件并且检查它的版本，如果 PI-ActiveView 没有安装或者版本号低于 Codebase 属性指向的安装包的版本号，浏览器便会安装 PI-ActiveView 从而显示 PDI 文件。

第三步，浏览器检查嵌入对象的 DisplayURL 属性，这个属性表明 PDI 文件的位置，如果第二步成功，那么控件便会下载 PDI 文件。

第四步，控件中 Acview.exe 启动并利用 PI-API 连接到 PI 数据库读取 PDI 文件中需要的数据。

经过以上四步，便实现了 PI 数据库的 WEB 开发，用户可以通过浏览器访问 PI 数据库。

由于 ActiveX 技术的安全性问题, 微软公司在 IE 浏览器中使用了更多的手段来限制 ActiveX 以保证其安全。如使用 IE6.0 SP2 及其后续版本访问包含有 ActiveX 控件的网页, 浏览器不会自动激活控件而是需要用户点击它们来激活。这样使用 ActiveX 技术的 PI-ActiveView 控件在每次网页刷新时都需要点击来激活, 这给用户在使用上造成了一定的不便。这个问题可以通过利用网页外部的脚本文件中的 Javascript 动态调用 PDI 文件来解决^[24], 这样 PI-ActiveView 的 PDI 文件就能够在加载时自动激活从而立即接受用户的输入。需要手动点击激活的网页代码 (HTML) 如下

```
<div id="SystemPdiGraph" align="center">
  <object classid="CLSID:4F26B906-2854-11D1-9597-00A0C931BFC8"
    width="90%" height="60%" codebase = "http://10.14.91.115/Cab">
    <param name="DisplayURL" value="http://10.14.91.115/CCADMSpdi/gt.pdi" />
  </object>
</div>
```

经过修改的自动激活 ActiveX 控件的等效代码如下分为两段, 第一段仍然为 HTML:

```
<div id="SystemPdiGraph" align="center">
  <script src="LoadPDI.js" type="text/javascript"></script>
  <script type="text/javascript">
    LoadPDI( "SystemPdiGraph",
      "CLSID:4F26B906-2854-11D1-9597-00A0C931BFC8",
      "GTPDI_ID", "90%", "60%",
      "http://10.14.91.115/CCADMSpdi/gt.pdi",
      "http://10.14.91.115/Cab")
  </script>
</div>
```

第二段为网页引用的脚本文件 (Javascript)

```
function LoadPDI(DivID, CLSID, ObjectID, WIDTH, HEIGHT, URL , Codebase)
{
  var d = document.getElementById(DivID);
  d.innerHTML =
    '<object classid=' + CLSID +
    ' id=' + ObjectID +
    ' width=' + WIDTH +
    ' height=' + HEIGHT +
    ' Codebase=' + Codebase +
    '><param name="DisplayURL" value=' + URL + '>' +
    '</object>';
}
```

在 PI-ActiveView 数据显示过程中, PDI 对象 Codebase 属性非常重要, 用户第一次浏览带有 PDI 对象的网页时, 浏览器将根据 Codebase 属性的值自动下载 PI-ActiveView 控件, 否则 PDI 文件不能显示。Codebase 属性指向的 PI-ActiveView

控件的安装包是一些 CAB 文件。PI-ActiveView 中提供了 CABWIZARD 工具来制作和发布在浏览器中自动安装 PI-ActiveView 控件的 CAB 文件。

3.2.3 PI 数据库应用程序接口 PI-API

利用 PI-ProcessBook 和 PI-ActiveView 等 PI 数据库的客户端工具可以非常简便的实现与数据库中的数据进行浏览和交互。但是如果需要实现真正的 PI 数据库的 WEB 开发以避免上一节所述的 PI-ActiveView 的缺点或者需要更进一步的在独立应用程序如半山发电厂联合循环机组辅助决策系统中读取和写入 PI 数据库,则需要使用 PI 数据库提供的编程接口 PI-API、PI-API 或者 PI-OLEDB。

PI 应用程序接口(PI-API)是一组实现编程存取 PI 数据库的函数库^[25]。PI-API 提供了在各种不同硬件和软件环境下通用的 PI 数据库接口,用来实现在应用程序中存取 PI 数据库的快照、存档和其他信息。这些平台包括 Windows、Unix 和 Linux 等。

PI-API 是基于 C/S 架构设计的。当客户端第一次调用 PI-API 函数时,客户端首先与数据库建立连接,如果服务器验证登录信息成功,客户端与 PI 服务器的会话便建立,下一次通过 PI-API 访问时便无须认证,直到应用程序断开连接或者网络出现故障。半山发电厂联合循环机组辅助决策系统的后台计算程序和 B/S 系统中都需要直接连接 PI 数据库,由于 PI-API 是针对桌面程序和 C/S 架构设计的,传统 WEB 程序编程模型例如 ASP 不支持直接调用 PI-API 提供的动态链接库中的函数。但是由于 .NET 框架对于桌面应用程和 WEB 程序提供了一致的编程模型,这使得在 ASP.NET 中同样使用 PI-API 来连接 PI 数据库成为可能。

PI-API 提供的函数格式与 Windows API 类似,函数的初始值和结构都通过参数传递,而函数的返回值则是代表函数的执行是否成功,通常 0 代表函数执行成功,其他值则对应具体的错误。依照函数的功能,PI-API 中的所有函数如表 3.1 被分为几组,并利用函数名的前缀表示函数的组,方便函数调用和查询。

表 3.1 PI-API 函数的分类

函数组名称	函数名前缀	描述
存档函数	piar_	存取和检查 PI 数据库中的存档数据
批函数	piba_	获取 PI 数据库中的批(Batch)设置和查询批活动
事件日志函数	piel_	查询和增加 PI 数据库和客户端节点的事件日志
登录函数	pilg_	提供 PI 数据库连接服务和常用 PI 任务用户对话框
消息日志函数	pilg_	发送消息到消息日志文件或者标准输出
点函数	pipt_	获取 PI 数据库中点的属性
快照函数	pipt_	存取和检查 PI 数据库中的快照数据
时间函数	pitm_	修改系统和服务器的时间或者时间格式
工具函数	piut_	提供 PI 数据库连接服务和获取 PI 数据库信息

当通过 PI-API 连接安装于 Windows 或者 Unix 平台的 PI 数据库时, 可以使用它提供的扩展 API 函数, 这些函数提供了更多的功能。扩展函数仍然按照功能被分到了各个组中。

PI-API 和 Windows API 相同, 是使用 C 语言编写, 在 .NET 框架程序设计中。某些情况下为了提高效率会避开 .NET 类库而直接调用 Windows API, 这样就要使用到 P/Invoke^[26]。同样在 .NET 框架中调用 PI-API 函数也要使用 P/Invoke。

P/Invoke 是 .NET 框架中公共语言运行时提供的的一个特性, 它允许从托管代码中调用动态链接库中的平台原生代码。使用 P/Invoke 时, CLR 调用 DLL 并把其中的非托管类型转换为 .NET 框架使用的通用类型系统支持的类型。但是调用的原生代码不受 .NET 框架中提供的类型安全和垃圾收集等特性的支持, 因此 .NET 框架并不提倡在能够使用 .NET 类库时去使用 P/Invoke。提供 P/Invoke 是为了让 .NET 框架可以通过调用原生代码来实现 .NET 框架类库没有提供的功能, 例如调用 PI-API; 或者为了提高程序性能, 例如在某些关键位置直接调用 Windows API。

由于不同的操作系统和 C 编译器对于 C 语言类型的长度解释有所不同, 因此在 PI-API 中定义了一组别名来消除这种差异, 别名通常为类型名称和长度, 例如 int32 和 float64 等。同时 PI-API 定义了 PI 指针类型 (PIPTR) 来消除不同平台指针的差异, 使用时需要制定指针的类型。使用 P/Invoke 进行调用时的关键是需要找到和使用与 PI-API 中的 C 语言类型不同但是兼容的 C# 支持 CLR 数据类型。

通过 PI-API 存取 PI 数据库时涉及到的点类型包括数值型、数字型和字符串型三种。PI-API 函数中, 数值型返回数值, 数字型返回数字或者数据库中已设置好的数字对应的描述字符串, 字符串型则通常返回字符串首地址和字符串大小。

对于 PI-API 中 int32 和 float64 这类的数值型的参数, 在 .NET 框架中存在直接对应 System.Int32 和 System.Double。如果参数为数值型 PI 指针, 则传递在 .NET 框架中对应的类型的引用 (通过 ref 或者 out 关键字)。在通过 PI-API 修改或者设置数据时, 数值型 PI 指针在某些情况下必须指向 null, 而在 .NET 框架中 System.Int32 和 System.Double 等值类型对应的引用都有对应的地址而不可能是 null, 因此如果数值型 PI 指针需要指向 null 时, 则它对应的是 .NET 框架类型为 object, 不指向 null 时把数值型 PI 指针对应的值装箱赋给 object 对象, 指向 null 时直接给 object 赋值 null。

PI-API 中对于数字型的数字通常用 istat 参数返回, istat 是一个 int32 型的 PI 指针, 指向数据即为数字型的数字, 然后通过 pipt_digstate 函数根据数字可以得到它的描述字符串。

由于 C 和 .NET 框架中字符串的结构完全不同, 并且存在于托管和非托管两个环境中, 因此字符串的存取又是利用 C# 调用 PI-API 的一个难点。对于字符串

类型, PI-API 的参数为 `bval` 和 `bsize`, 前者是指向字符串首地址的不透明 PI 指针 (`void PIPTR *`), 即该指针没有类型, 对应于 .NET 框架类型是 `System.IntPtr`, 因为它本身就表示一个指针, 所以使用该类型时不需要使用关键字 `ref` 或者 `out`。后者是表示字符串的长度 `uint32` 类型的 PI 指针。通过这两个参数表可以读取或者修改字符串。

读取字符串时, 首先使用 `Marshal.AllocHGlobal` 函数为 `System.IntPtr` 分配一块非托管指针内存, 然后把这个参数传递到 PI-API 函数得到返回值; 其次把返回的 `System.UInt32` 类型表示的字符串长度转换为 `System.Int32` 类型; 接着使用 `Marshal.PtrToStringAnsi` (`System.IntPtr`, `Int32`) 分配托管 `String`, 然后从 `System.IntPtr` 指向的非托管 ANSI 字符串向其复制指定数目的字符, 这样就得到了 NET 框架中 `String` 类型的托管字符串; 最后由于我们在非托管环境中分配了内存, 而 .NET 框架中对这一块区域是不能进行垃圾回收, 这就需要使用 `Marshal.FreeHGlobal` 函数来释放非托管字符串占用的空间以避免内存泄漏。

设置字符串时, 同样首先声明 `System.IntPtr` 对象; 其次使用 `Marshal.StringToHGlobalAnsi` 函数把托管字符串复制到非托管内存, 并把非托管字符串的首地址赋值给 `System.IntPtr`; 接着可以通过 `String.Length` 方法得到字符串的长度赋给 `bsize`, 这一步也可以把 `bsize` 设置为 0, 这样 PI-API 会使用 C 语言的 `strlen` 函数测量非托管字符串的长度。然后函数调用时使用 `System.IntPtr` 对象为参数, 对应不透明 PI 指针。最后使用 `Marshal.FreeHGlobal` 函数释放非托管字符串占用的空间。

在 PI-API 的函数参数中还有一种常见参数类型为 `PITIMESTAMP`, 这是 PI-API 定义的用于表示时间的结构类型。其 C 语言原型为

```
typedef struct _PITIMESTAMP {  
    int32 month; /* 1-12 */  
    int32 year; /* four digit */  
    int32 day; /* 1-31 */  
    int32 hour; /* 0-23 */  
    int32 minute; /* 0-59 */  
    int32 tzinfo; /* timezone information */  
    double second; /* 0-59.99999999.... */  
} PITIMESTAMP;
```

因为结构体中类型均为上面介绍过的数值型, 所以使用 C# 定义一个相同的结构体比较简单

```
public struct PITIMESTAMP{  
    public Int32 month; /* 1-12 */  
    public Int32 year; /* four digit */  
    public Int32 day; /* 1-31 */  
    public Int32 hour; /* 0-23 */  
    public Int32 minute; /* 0-59 */
```

```

public Int32 tzinfo; /* timezone information */
public Double second; /* 0-59.99999999.... */
}

```

因为结构体为值类型，如果参数为 PITIMESTAMP 类型 PI 指针，则传递在 .NET 框架中定义的结构体对象的引用。

半山发电厂联合循环机组辅助决策系统中主要使用 PI-API 来直接读取或者设置 PI 数据库中的数据，一个比较具有代表性并且比较复杂的函数是用于读取快照数据的扩展函数 `piar_getsnapshotx`，下面给出使用 C# 调用该函数的对应方法定义。`piar_getsnapshotx` 的 C 语言声明为：

```

int32 PIPROC pism_getsnapshotx(
    int32 ptnum,
    float64 PIPTR *drval,
    int32 PIPTR *ival,
    void PIPTR *bval,
    uint32 PIPTR *bsize,
    int32 PIPTR *istat,
    int16 PIPTR *flags,
    PITIMESTAMP PIPTR *timestamp
);

```

使用 P/Invoke 时首先使用 C# 对它进行声明，函数参数根据上述规则转换为 CLR 支持的类型，之后便可在需要时调用它。C# 声明代码如下

```

public static extern Int32 pism_getsnapshotx(
    Int32 ptnum,
    ref double objval,
    ref Int32 ival,
    System.IntPtr bval,
    ref UInt32 bsize,
    ref Int32 istat,
    ref Int16 flags,
    ref PITIMESTAMP timestamp
);

```

使用 C# 调用 PI-API 时，由于每个类需要进行 P/Invoke 时都必须先声明使用的 PI-API 函数，并且简化读取和写入过程。半山发电厂联合循环机组辅助决策系统对 PI-API 进行了简单的封装。这样只需要使用如下语句就可以实现读取快照数据。

```

PISnapshot.GetValueex(String tagName, DateTime time).sValue

```

3.2.4 PI 数据库软件开发工具 PI-SDK

PI 数据库软件开发工具（PI-SDK）是使用面向对象思想和技术开发 PI 数据库的编程接口，主要由一组 ActiveX 控件和类库组成。与 PI-API 相比，它提供了更多的特性。如提供 PI 数据库的面向对象编程模型；支持 PI3 系统的高精度

时间、用户自定义属性等新特性；提供一组 ActiveX 控件来完成如用户登录、Tag 查找等常用功能；支持多线程和异常处理等。

PI-SDK 由表示 PI 数据库的一组有层次的对象集合构成^[27]，如图 3.7。应用程序首先获得位于最高层次的对象，然后在通过它的属性或者方法来获得其他对象或对象集合。

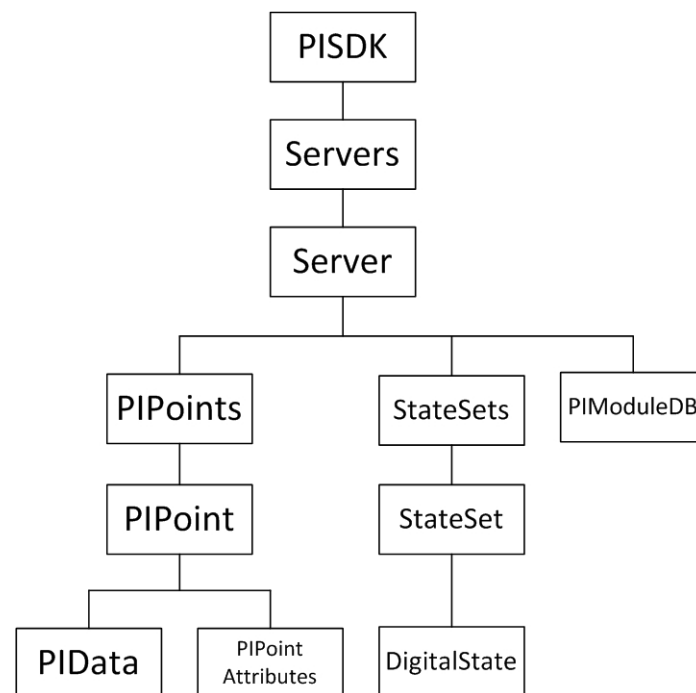


图 3.7 PI-SDK 层次对象模型

位于层次对象模型顶端的是 PISDK 对象，它的 Servers 属性返回 PI 服务器对象集合 (Servers)，通过 Servers 对象可以得到保存在已知服务器表中的服务器对象 (Server)。

服务器对象表示了一个独立的 PI 数据库服务器。使用 PI-SDK 存储数据首先要建立和服务器进行连接。初始化连接可以通过 Server 对象的 Open 方法显式的进行，也可以在第一次调用 Server 对象属性或者方法的时候 PI-SDK 隐式的进行。通过 Server 对象的属性将得到许多常用对象。

通过 Server 对象返回的第一类常用对象是点集合 (PIPoints) 和点 (PIPoint)，PIPoint 对象表示 PI 数据库中的点，这些点可能是通过传感器测量得到的直接测点，或者是经过计算转换的其他点。通过 PIPoint 可以读取和修改点的历史数据和存档数据，也可以读取和修改点的属性。

通过 Server 对象返回的第二类常用对象是状态集集合 (StateSets)、状态集 (StateSet) 和数字状态 (DigitalState)，他们都与 PI 数据库中的开关量 (Digital) 类型有关。

DigitalState 对象用来表示 PIPoint 对象的状态，包括设定状态和错误状态，

设定状态是用户在 StateSet 中定义的, 只能用于 PI 数据库中 Digital 类型的点; 错误状态是系统定义的“System” StateSet 的成员, 可以用于任何类型的点, 用于在点的数据出现错误时返回点的错误信息。DigitalState 对象的两个主要属性为 Code 和 Name。Code 是标示状态的整型数字, 主要用于传输。Name 是设置好的与 Code 对应的描述状态的字符串, 主要用于显示。

通过 Server 对象返回的第三类常用对象是模块化数据库对象(PIModuleDB)等, 他们是 PI3 系统中的模块化数据库等新特性的模型, 在半山发电厂联合循环机组辅助决策系统的开发中没有使用。

由于 PI 数据库中, 点的数据是与时间标签对应的, 因此 PI-SDK 把这个特点抽象为 PIValue 对象, PIValue 通过它的 TimeStamp 属性返回时间标签, 通过 Value 属性返回实际数据。

PIData 对象是 PI-SDK 层次模型的一个关键对象, 读取和写入数据都通过这个对象实现。PIData 的 Snapshot 属性返回点的实时数据、ArcValue 方法返回点的历史数据, 两者均返回 PIValue 对象。写入数据到 PI 数据库则通过 UpdateValue 方法实现。

上述是半山发电厂联合循环机组辅助决策系统开发中主要使用的 PI-SDK 对象, 它们提供了一个面向对象的模型来读取、修改 PI 数据库中的数据以及实现其他功能。

PI-SDK 基于组件对象模型 (COM), COM 是微软公司提出的上一代编程模型, 其本质是用语言中立的方式执行对象, 这样可以使对象在不同的环境中创建和使用。理论上 COM 适用于不同的操作系统, 实际上它主要应用在 Windows 平台中, 因此 PI-SDK 主要用于开发 Windows 应用程序。

组件对象模型与 .NET 框架均是微软提出的编程模型并且目标相似, 但是他们有着本质上的区别。CLR 程序是托管执行的, 其中包含内存回收器和通用编程模型等。COM 是非托管执行的、它使用引用计数, 虽然有一个二进制标准, 但是存在许多不同的编程模型。

为了实现传统的 COM 程序与 .NET 程序之间的相互调用, .NET 提供了包装类 RCW(Runtime Callable Wrapper)和 CCW(COM Callable Wrapper)^{[28][29]}。RCW 获取 COM 组件并将其包装起来, 这样在 NET 客户端能够像调用普通 .NET 对象一样调用它; CCW 则用来包装 .NET 对象供 COM 客户端使用, 如图 3.8。

使用 .NET 框架调用 PI-SDK 对 PI 数据库进行开发时使用了 RCW, 它处理并掩盖 COM 到 .NET 之间的转换, 由于 RCW 自动进行数据的封送处理, 因此不需要像调用 PI-API 时进行数据封送, 简化了开发。此外它还解决了对象的生存期和错误处理等问题。RCW 是由 CLR 根据 Interop 程序集中包含的元数据信息在运行时动态创建的, 因此使用前需要生成 COM 组件的 Interop 程序集, 它可以使用 .NET 框架提供的工具生成, OSI 公司已经在 PI-SDK 中提供了生成好的

Interop 程序集。

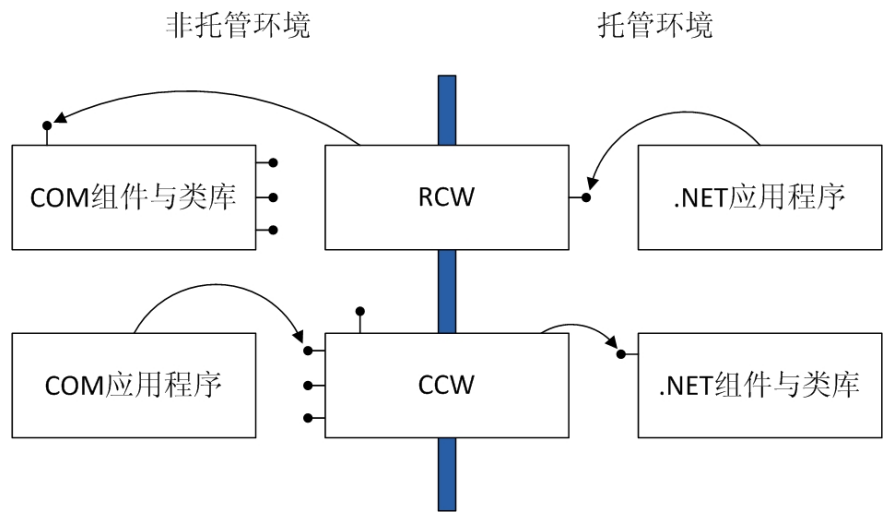


图 3.8 .NET 框架与 COM 模型的互相调用

实际开发中，使用 C#语言调用 PI-SDK 存取数据库中数据很方便，首先需要添加对 PI-SDK 的引用，实现读取实时快照数据为例的代码如下：

```
PISDK.IPISDK pisdsk = new PISDK.PISDKClass();
PISDK.Server srv;
PISDK.PIPoint point;
PISDK.PIValue pival;
srv = (PISDK.Server)pisdsk.Servers["LabPI"];
srv.Open("UID=piadmin;PWD=");
point = srv.PIPoints["cdep158"];
pival = point.Data.Snapshot;
```

在使用隐式登录时，可以简化如下

```
PISDK.IPISDK pisdsk = new PISDK.PISDKClass();
pisdsk.Servers["LabPI"].PIPoints["cdep158"].Data.Snapshot
```

除开 PI-API 和 PI-SDK，PI 数据库还提供了 PI-OLEDB，通过它应用程序能够像访问普通关系数据库一样访问 PI 系统，由于半山发电厂的 PI 数据库中没有该组件，因此半山发电厂联合循环机组辅助决策系统将结合 PI-API 和 PI-SDK 存取 PI 数据库中的数据。

3.3 对象关系映射 ORM

面向对象编程思想和关系数据库都是当前应用程序开发中的主流技术。面向对象是从软件工程基本原则发展起来的，而关系数据库则是从数学理论发展而来的，两套理论存在显著的区别。为了解决这个不匹配的现象便出现对象关系映射技术。

对象关系映射（Object Relational Mapping），简称 ORM，它不是一种具体的

产品，而是映射、转换面向对象程序设计语言和关系数据库中不同类型系统的技术。

面向对象编程思想中使用类、多态、继承等特性，一定程度上简化了程序开发过程，提高了了程度的可读性。关系数据库系统则在大量数据查找和排序、数据并发访问和事务等方面具有优势。基于 ORM 框架完成映射后，既可利用面向对象程序设计语言的简单易用性，又可利用关系数据库的数据操作优势。

ORM 是一个高层的框架，在具体存取关系数据库时，不同的语言使用不同的方法，因此每种 ORM 产品一般只能应用于某种语言。但同时因为 ORM 框架架构和语言是无关的，所以在数据存储部分稍作修改就能够应用于不同程序设计语言。例如面向 Java 的 ORM 框架 iBatis 和面向.NET 的 ORM 框架 iBatis.NET 实质上是相同的。

由于 ORM 技术具有不少优点，简化了基于关系数据库应用程序的开发和维护，出现了各种商业的或者开源的 ORM 产品，常见的应用于.NET 平台的开源产品有 NHibernate、iBatis.NET 和.netTiers 等。由于这些框架在实现时考虑了产品通用性，一般比较复杂，并且针对性不够强，因此在半山发电厂联合循环机组辅助决策系统中，将不使用现成的框架而是独立实现对象关系映射。

半山发电厂联合循环机组辅助决策系统的对象关系映射主要实现数据表中数据的 CRUD（创建、读取、更新和删除），这些是关键和主要使用到的功能，而对于表之间的关系则尽可能的简化和不映射，因为他们不常用并且实现复杂。

负责 ORM 的命名空间为 SQLServerModel 和 SQLServerDAL。SQLServerModel 命名空间用于将数据库映射为对象，数据库的数据表对应命名空间中的类，而数据表中的字段则对应类的属性。

半山发电厂联合循环机组辅助决策系统的维修间隔计算模块使用数据表 maintaince_StartStop 来统计每次的启停数据。其结构如下

表 3.2 maintaince_StartStop 数据表结构

字段名	类型	说明
StartStopID	int	ID
UnitNumber	int	机组
FlameOnTime	datetime	点火时间
FlameOffTime	datetime	熄火时间
FiredHours	float	点火小时数
MaxPower	float	最大功率
IsTripped	bit	是否跳闸
TrippedPower	float	跳闸功率
StopedHours	float	上次停机时间
PowerHours	float	出力小时数

则它在 `SQLServerModel` 命名空间中对应的映射类为 `StartStopModel`，每个字段则是 `StartStopModel` 的一个属性，如图 3.9。



图 3.9 `StartStopModel` 类和 `StartStopDAL` 类

`SQLServerDAL` 命名空间的类则用于把 `Model` 对象转化成 SQL 语句返回数据库执行和把数据库返回的数据转换成为对应的 `Model` 对象。`maintaince_StartStop` 数据表在该命名空间中对应的数据存取类为 `StartStopDAL`，如图 3.9，它提供一系列方法完成 CRUD，从而实现对数据库中数据的操作。

`StartStopDAL` 类的每个方法存储数据库使用 .NET 框架提供的 ADO.NET 技术实现。下面是 `GetStartStops` 和 `Update` 方法和 `GetStartStops` 方法的实现代码。前者用于将对象更新到关系数据库，后者读取数据库中的数据到对象。

```
public void Update(StartStopModel model)
```

```

{
    StringBuilder sbSqlUpdate = new StringBuilder();
    sbSqlUpdate.Append("update maintaince_StartStop set ");
    sbSqlUpdate.Append("UnitNumber=@UnitNumber,");
    ...
    sbSqlUpdate.Append("PowerHours=@PowerHours");
    sbSqlUpdate.Append(" where StartStopID=@StartStopID");
    SqlParameter[] parameters = {
        new SqlParameter("@UnitNumber", SqlDbType.Int,4),
        ...
        new SqlParameter("@PowerHours", SqlDbType.Float,8),
        new SqlParameter("@StartStopID", SqlDbType.Int,4));
    parameters[0].Value = model.UnitNumber;
    ...
    parameters[8].Value = model.PowerHours;
    parameters[9].Value = model.StartStopID;
    SqlHelper.ExecuteNonQuery(SqlHelper.ConnectionString, CommandType.Text,
sbSqlUpdate.ToString(), parameters);
}

public List<StartStopModel> GetStartStops (int unitNumber)
{
    List<StartStopModel> rtnModels = new List<StartStopModel>();
    StartStopModel model;
    StringBuilder sbSqlSelect = new StringBuilder();
    sbSqlSelect.Append("select * from maintaince_StartStop ");
    sbSqlSelect.Append(" where UnitNumber=@UnitNumber");
    SqlParameter[] param = {
        new SqlParameter("@UnitNumber", SqlDbType.Int,4));
    param[0].Value = unitNumber;
    using (SqlDataReader rdr = SqlHelper.ExecuteReader(SqlHelper.ConnectionString,
CommandType.Text, sbSqlSelect.ToString(), param))
    {
        if (rdr.HasRows)
        {
            while (rdr.Read())
            {
                model = new StartStopModel(rdr.GetInt32(0), rdr.GetInt32(1),
rdr.GetDateTime(2),rdr.GetDateTime(3),rdr.GetDouble(4),rdr.GetDouble(5),rdr.GetBoolean(
6), rdr.GetDouble(7), rdr.GetDouble(8),rdr.GetDouble(9));
                rtnModels.Add(model);
            }
        }
        else
        {
            rtnModels = null;
        }
    }
    return rtnModels;
}

```



```
}
```

完成对象关系映射以后，在程序中操作数据将以面向对象的思想来完成，无论是在后台服务器端应用程序还是在 WEB 服务器应用程序中，如果需要添加 maintince_StartStop 数据表的数据。只需要首先声明 StartStopDAL 和 StartStopModel 对象，然后给 StartStopModel 对象的属性赋值，最后通过 StartStopDAL 的 Insert 方法将数据送到数据库，代码如下：

```
StartStopModel startStopModel = new StartStopModel();
StartStopDAL startStopDal = new StartStopDAL();
startStopModel.UnitNumber = [数据];
...
startStopModel.IsTripped = [数据];
startStopModel.TrippedPower = [数据];
startStopDal.Insert(startStopModel);
```

目前半山发电厂联合循环机组辅助决策系统 ORM 中的数据库查询方面还不是很完善，不能让用户自定义条件来进行查询，这些还需要在后续维护中进行完善。

3.4 异步 JavaScript 和 XML

3.4.1 Ajax 概述

半山发电厂联合循环机组辅助决策系统主要部分采用 B/S 架构，即用户通过浏览器访问位于服务器上的 WEB 应用程序。

在传统的 WEB 程序中，用户通过浏览器访问服务器，用户提交操作时浏览器就向 Web 服务器端发送一个请求。服务器接收并处理传来的请求，最后将处理过的数据返回到客户端，因为在客户端传送到服务器和服务器返回客户端的数据中有很大部分是描述页面的 HTML，并且大部分 HTML 代码往往是相同的，所以这种做法浪费了许多带宽。同时由于每次应用的交互都需要服务器发送请求并刷新整个页面，所以应用的响应时间依赖于服务器的响应时间，这导致了用户界面的响应比本地应用慢得多。

异步 JavaScript 和 XML（Asynchronous JavaScript and XML，简称 Ajax）技术的出现部分解决了这个问题。

Ajax 提供与服务器异步通信的能力^[31]，如图 3.10 所示，从而使用户不必等待请求/响应循环。借助于 Ajax，可以在用户单击按钮提交操作时，浏览器使用 JavaScript 和 DHTML 立即更新用户界面，并向服务器发出异步请求，以执行更新或查询数据库。当请求返回时，同样使用 JavaScript 和 CSS 来相应地更新用户界面，而不是刷新整个页面。使用 Ajax 技术，服务器和客户端交互时只需要传输关键数据，而大部分没有改变的 HTML 代码并没有在它们之间传输，从而节

省了带宽，能够更快的看到服务器响应的结果。同时很多的处理工作可以在发出请求的客户端机器上完成，所以 Web 服务器的处理时间也减少了。这样，用户几乎不知道浏览器正在与服务器通信，Web 站点看起来是即时响应的，就像是在运行本地应用程序一样。

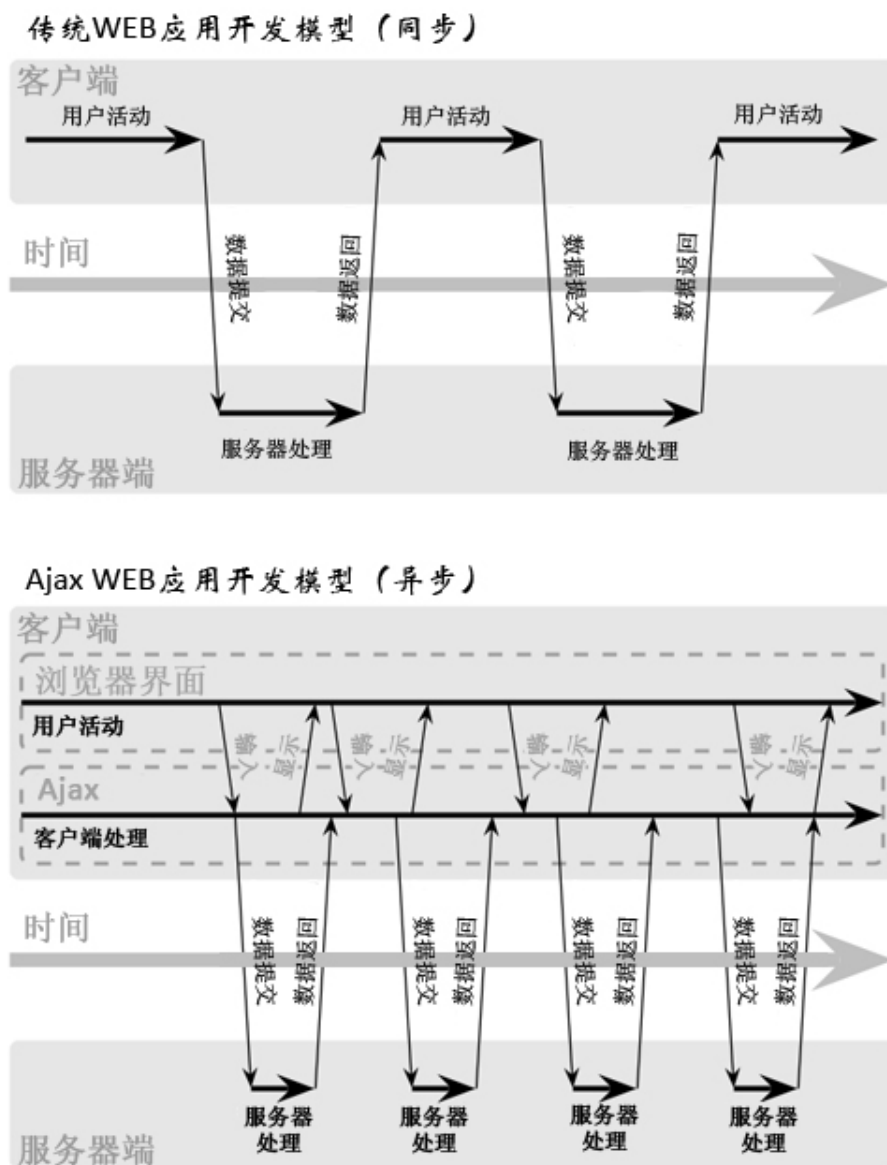


图 3.10 传统与 Ajax WEB 应用开发模型比较

综上，使用 Ajax 的最大优点是能够在不更新整个页面的情况下进行数据传输和维护，并且避免发送重复信息，使得 WEB 应用程序更为迅捷地响应用户操作。

实质上 Ajax 不是一种新技术，而是有机的把一系列原有技术如 Javascript、XHTML 和 CSS 等结合起来，从而实现数据的异步传输。因为这些原有技术在现代浏览器上都受到了很好的支持，因此访问 Ajax 应用不需要安装任何浏览器插

件，这也是它的一个主要优点。

Ajax 应用有三个主要缺点，首先它破坏了浏览器的部分正常功能，如浏览器的后退按钮不能正常返回上一个操作界面、使用动态页面更新使用户难于保存特定的状态的网页。其次，如果存在比较大的网络延迟时，用户通过浏览器提交操作后，由于数据在后台传输导致前台没有响应，仿佛浏览器死机一样。

最后一个缺点则是 Ajax 应用的开发比较复杂。虽然 Ajax 可不使用任何插件运行于各种浏览器中，但不同的浏览器对于 Ajax 使用的 JavaScript/XHTML/CSS 技术的支持总有部分不太相同或是有 Bug，甚至同一浏览器的不同版本间也有可能不一样，这就导致开发过程中必须了解不同浏览器对这些技术的支持差异并测试应用程序的浏览器兼容性，增加了开发难度。此外，如果同时有多个 Ajax 请求发送到服务器端，最先返回的结果不一定对应最先提交的请求。

上述缺点可以通过使用一些 Ajax 框架得到弥补，目前 Ajax 技术发展很快，也有很多优秀的 Ajax 框架出现，如 Prototype、Dojo 和 jQuery 等。在半山发电厂联合循环机组辅助决策系统中考虑到应用 Ajax 技术可以很大程度上改进用户体验，所以已采用异步 JavaScript 和 XML 技术。为了降低开发难度，使用微软公司发布的 ASP.NET AJAX 框架辅助开发。

3.4.2 ASP.NET AJAX 框架

ASP.NET AJAX 框架提供了一组跨浏览器的客户端脚本库，从而避开了浏览器的不一致性问题，同时它还包括一组服务器端组件并且和 ASP.NET 2.0 服务器端开发框架能够很好的集成在一起。

ASP.NET AJAX 服务器端组件包括 ASP.NET 服务器控件和类库，用来管理用户界面、程序流和控件扩展等。

服务器端控件包括 ScriptManager、UpdatePanel、UpdateProgress 和 Timer。ScriptManager 控件是 ASP.NET AJAX 框架的基础，使用其他控件必须先使用它，他用于管理客户端脚本资源、实现局部回调和用户自定义脚本等。UpdatePanel 控件则是核心控件，它允许只刷新页面的指定部分，即使原先需要全屏刷新才能产生效果的 ASP.NET 服务器控件在异步调用不全屏刷新使产生效果，自动完成 Ajax 处理。UpdateProgress 用于在页面异步刷新时显示信息，避免在长时间回调时给用户浏览器死机的假象。Timer 控件则用于定时向服务器提交请求。

同时 ASP.NET AJAX 服务器端组件还包含了对 ASP.NET WEB 服务的扩展。在添加 ASP.NET AJAX 的引用后，只要给 WEB 服务指定 ScriptService 属性，便可以在客户端简单的实现 Javascript 对 WEB 服务的调用。

ASP.NET AJAX 客户端脚本库是对 Javascript 的扩展，通过这个库可以在

Javascript 开发中使用面向对象的特性,从而使客户端编程和服务端编程具备一致性。跨浏览器层提供 ASP.NET AJAX 脚本的浏览器兼容性支持;核心服务实现对 Javascript 的面向对象特性的扩展,使之支持类、事件处理和继承的支持;基类库实现 stringBuilder 等客户端组件和扩展错误处理支持;此外客户端脚本库还支持调式模式等其他特性。

半山发电厂联合循环机组辅助决策系统系统主要使用服务器端 UpdatePanel 控件实现常规 ASP.NET 服务器控件局部刷新和使用客户端脚本库简化 Javascript 调用 WEB 服务,从而数据异步传输,提高程序效率和增强用户体验。

半山发电厂联合循环机组辅助决策系统的首页实现无刷新导航,即用户点击菜单后,在内容区域异步加载菜单对应的 ASP.NET 用户控件。实现过程包括三个重要方面,第一是动态加载用户控件,第二是利用 UpdatePanel 实现局部刷新加载。第三是保持用户控件状态。

动态加载用户控件首先需要清除 Placeholder 里面已经存在的控件,然后使用 TemplateControl 的 LoadControl 静态方法加载指定路径的用户控件,加载以后一般需要指定控件的 ID 属性为任意值,在辅助决策系统中将其制定为控件控件名,否则可能在控件转换时出现错误。加载函数代码如下:

```
public Control LoadWebUserControl(string userControlName)
{
    ControlType = userControlName;
    Placeholder1.Controls.Clear();
    Control ctl = LoadControl(userControlName);
    ctl.ID = userControlName.Substring(userControlName.LastIndexOf('\\') + 1,
        userControlName.LastIndexOf('.') - userControlName.LastIndexOf('\\') - 1);
    Placeholder1.Controls.Add(ctl);
    return ctl;
}
```

在得到返回的 Control 对象以后,再把它转换为特定用户控件的类型,例如加载统计热通道维修间隔计算结果的 wucHGPMaintainceCalc 控件

```
wucHGPMaintainceCalc hgp =
(wucHGPMaintainceCalc)LoadWebUserControl(@"UserControls\Maintaince\wucHGPMaint
ainceCalc.ascx");
```

为了实现异步回调、局部刷新加载用户控件,需要在页面中使用 UpdatePanel,并设置 UpdatePanel 的异步回调触发事件属性为导航菜单中按钮的点击事件,这样在用户点击导航菜单切换页面时,位于 UpdatePanel 的 ContentTemplate 中 Placeholder 控件在默认情况下将局部刷新加载用户控件。UpdatePanel 的实现过程自动应用到 Ajax WEB 模型。

```
<asp:UpdatePanel ID="UpdatePanelMain" runat="server">
    <Triggers>
        <asp:AsyncPostBackTrigger ControlID="lbtnUnit1" EventName="Click" />
    </Triggers>
</asp:UpdatePanel>
```

```

...
<asp:AsyncPostBackTrigger ControlID="lbPerformanceGTSystemGraph"
EventName="Click" />
<asp:AsyncPostBackTrigger ControlID="lbPerformanceBoilerSystemGraph"
EventName="Click" />
...
<asp:AsyncPostBackTrigger ControlID="lbMaintainceRoter" EventName="Click" />
</Triggers>
<ContentTemplate>
<asp:PlaceHolder ID="PlaceHolder1" runat="server"></asp:PlaceHolder>
</ContentTemplate>
</asp:UpdatePanel>

```

通过以上两部便实现了无刷新动态加载用户控件,但是此时如果点击用户控件中的按钮提交请求时,用户控件会消失或者不能保持控件的状态,这是由于 HTTP 协议的无状态特性决定的。这就还需要在提交时保持用户控件的状态,这在 ASP.NET 中可以使用 ViewState 实现。ViewState 在页面提交前保存其中信息发送回服务器端,并随服务器的返回数据一起返回。其实现如下:

```

protected string ControlType{
    get {
        if (ViewState["ControlType"] != null) {
            return ViewState["ControlType"].ToString();
        }
        return "";
    }
    set {
        ViewState["ControlType"] = value;
    }
}

```

首先在加载用户控件函数中保存用户控件名;

```
ControlType = userControlName;
```

然后在页面每次加载时重新加载控件

```

protected void Page_Load(object sender, EventArgs e){
    if (!string.IsNullOrEmpty(ControlType)) {
        LoadWebUserControl(ControlType);
    }
}

```

通过 ViewState 便实现了半山发电厂联合循环机组辅助决策系统首页的无刷新导航。

ASP.NET AJAX 框架在半山发电厂联合循环机组辅助决策系统中的另外一个主要应用是通过它的客户端脚本库对 Javascript 的扩展在客户端调用 WEB 服务器实现身份认证或者获取其他从服务器返回的信息。使用 Javascript 调用身份验证 WEB 服务的实例已在 3.1.3 节中给出,此处不在赘述。

除开上述两个主要应用之外,使用 ASP.NET AJAX 还可以实现简化 javascript

编程。例如传统的 Javascript 事件处理模型存在一个主要缺点即每个元素只允许有一个事件处理函数，但在半山发电厂联合循环机组辅助决策系统中，通常在事件发生时触发多个函数，例如按钮被点击时，浏览器一方面向服务器发送请求，另外一方面改变界面响应用户的操作，在不使用 Ajax 库时需要自己编写代码实现监听器模式将多个函数绑定到一个事件。通常，Ajax 库都提供了监听器模式来简化开发，例如 ASP.NET AJAX 直接提供了 AddHandler 函数实现弥补传统事件处理模型不支持多个事件处理的缺点。

第四章 辅助决策系统的架构和实现

4.1 辅助决策系统的架构

4.1.1 半山发电厂 SIS 系统

半山发电厂联合循环机组辅助决策系统是半山发电厂 SIS 系统的一部份。半山发电厂数据中心现有三台服务器，分别是 PI 服务器、WEB 服务器和应用程序服务器，皆安装 Windows 2003 Enterprise Edition 操作系统。PI 服务器是 SIS 系统的核心部分，安装 PI 实时数据库，并且 PI 服务器与 DCS 系统的接口机连接，接收接口机采集的机组运行数据并将其保存在实时数据库中，应用程序服务器安装 SQL Server 数据库和其他辅助机组优化运行的程序，如转子故障诊断系统等。WEB 服务器为半山已有 B/S 程序的服务器端。半山发电厂 SIS 系统结构图见 4.1

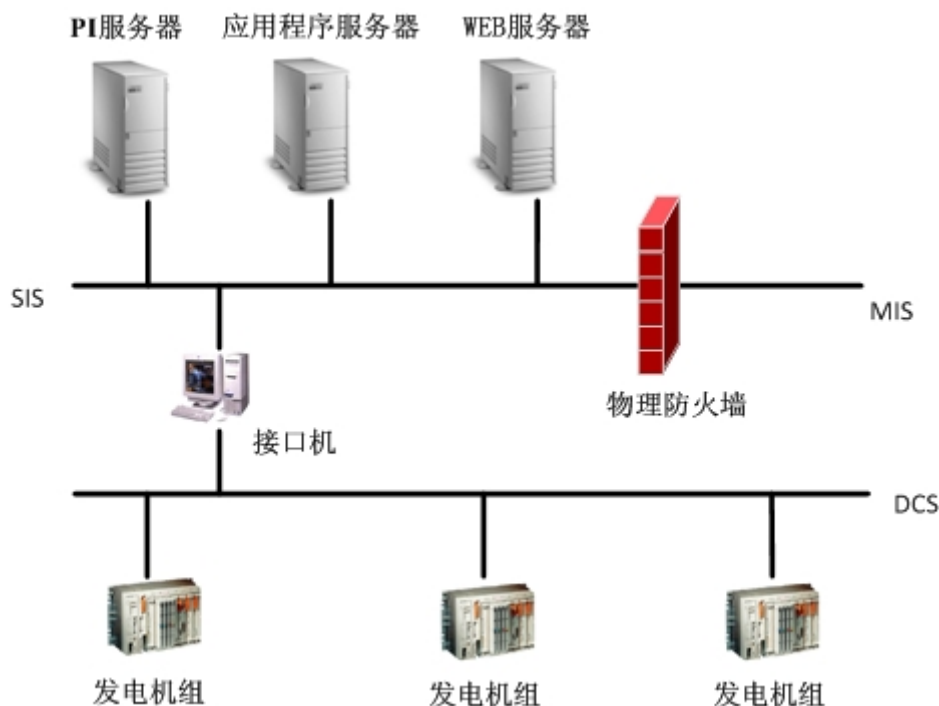


图 4.1 半山电厂 SIS 系统结构图

4.1.2 辅助决策系统架构模式的选择

目前主流的软件架构模式主要是 C/S 模式和 B/S 模式两种，C/S 架构软件(即客户机/服务器模式)分为客户机和服务器两层，客户机不是毫无运算能力的输入、输出设备，而是据有了一定的数据处理和数据存储能力，通过把应用程序的计算和数据合理地分配在客户机和服务器两端，可以有效地降低网络通信量和服务器

运算量。由于服务器连接个数和数据通信量的限制,这种结构的软件适于在用户数目不多的局域网内使用。**B/S**(浏览器/服务器模式)是随着 **Internet** 技术的兴起,对 **C/S** 结构的一种改进。在这种结构下,软件应用的业务逻辑完全在应用服务器端实现,用户表现完全在 **Web** 服务器实现,客户端只需要浏览器即可进行业务处理,是一种新的软件系统架构技术。

下面就两种架构模式在应用于电厂在线性能计算时的优缺点进行比较。

1. 硬件的投资成本:对于 **C/S** 架构模式,因为客户端可以承担一定的数据处理和数据存储工作,因此对于服务器的要求比 **B/S** 模式低。

2. 软件的开发难度:**B/S** 架构模式将 **C/S** 架构模式的两层结构发展到了三层结构,一般可以认为是在原有的 **Client** 层和 **Server** 层之间加入了 **Application Server** 层(也称为中间层)。**Application Server** 层承担了原来 **C/S** 架构模式中 **Client** 层和 **Server** 层的部分任务,这样使得 **Client** 层和 **Server** 层所承担的任务相对减轻。**Client** 层变成比较统一的界面,主要负责用户接口的实现,**Server** 层主要处理信息数据的存储和管理任务,**Application Server** 层负责具体数据的处理任务,而且可以根据处理任务的变化而变化。因为架构增加了一层,所以整个软件的架构规划的重要性的工作量增加。如果软件架构规划得当,各层之间的功能分割较好,耦合度较低,那么软件后期的编码阶段的开发难度和工作量将会减小。反之则开发难度和工作量增加。

3. 软件的易用度:使用 **B/S** 架构模式开发的程序,客户端是通过浏览器访问,对于使用客户只需要进行少量的培训,而 **C/S** 模式开发的软件,客户端可能有所不同,培训量较大。但是 **C/S** 架构模式开发的客户端,在用户界面的选择上比较多,功能也比较丰富,用户体验比 **B/S** 模式好。

4. 系统功能:**C/S** 开发的系统功能强大、交互能力强、系统运行效率高。相比 **B/S** 模式,由于 **C/S** 在逻辑结构上比 **B/S** 少一层,对于相同的任务,**C/S** 完成的速度总比 **B/S** 快,这就使得 **C/S** 模式更利于处理大量数据,因而适合于电厂在线性能实时计算软件的开发。但是 **C/S** 开发的系统结构复杂并且相对比较封闭。

5. 系统的维护与升级:在这个方面,**B/S** 架构模式明显体现着更为方便的特性。对一个稍微大一点得系统来说,系统管理人员如果维护 **C/S** 软件需要在几百甚至上千部电脑之间来回奔跑,效率和工作量是可想而知的,但 **B/S** 架构的软件只需要管理服务器就行了,所有的客户端只是浏览器,根本不需要做任何维护。无论用户的规模有多大,有多少分支机构都不会增加任何维护升级的工作量,所有的操作只需要针对服务器进行。当然针对这个缺陷,微软提出了 **clickonce** 技术已用来简化 **C/S** 程序的维护与升级,但是这种技术目前还没有普及。

综上,在开发在线性能计算软件的时候,两种模式各有利弊,但是考虑到系统维护和升级的便利,**B/S** 模式体现出了更大的优越性。因此本系统将主要采用

B/S 模式进行开发。部分无需人机交互的实时计算程序将以应用程序的方式后台运行在 SIS 服务器上。

4.1.3 服务器端计算程序架构

服务器端计算程序位于运行半山电厂的应用程序服务器上，负责与 PI 实时数据库和 SQL Server 数据库进行交互。服务器计算程序只需要一次性设置好它的相关选项后便无需用户的再次干预。他的主要作用是根据用户设置的计算间隔，从 PI 实时数据库中获取原始数据，根据维修间隔计算或者性能计算方法和模型，将计算后的二次数据存放到 PI 实时数据库或者 SQL Server 数据库中，方便 WEB 服务器的读取和使用，而不需每次用户通过 WEB 查看数据的时候重复进行计算。他的工作原理如图 4.2

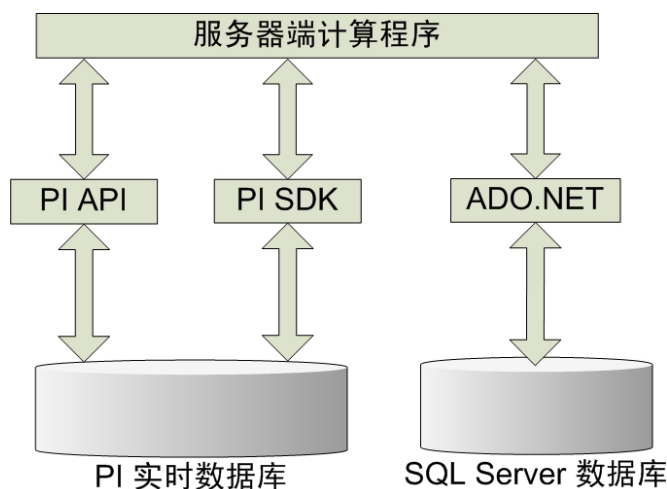


图 4.2 服务器端计算程序架构

4.1.4 WEB 程序架构

WEB 程序运行在半山电厂的 WEB 服务器上；客户端用浏览器对服务器进行访问，因为 WEB 程序的开发中使用了如 PI-ActiveView 和 OWC 等 ActiveX 控件，推荐使用 IE7.0 进行访问，最低要求为 IE6.0。

WEB 程序主要负责系统中需要人机对话的部分，如计算结果的显示、响应用户的输入和部分系统管理设置等。程序主要从 SQL Server 数据库中获得数据，并将结果存放进 SQL Server 数据库。部分实时数据与 PI 实时数据库进行交互。

使用 B/S 架构模式需要一个 WEB 服务器，为了与操作系统进行最佳的搭配，系统选用微软的 BackOffice 系列产品，以他的 IIS6.0 作为 WEB 服务器平台。IIS6.0 是集成于 Windows 2003 的一组 Web Server 软件，在安装了 Windows2003 操作系统后，可以通过添加删除程序添加 IIS。

联合循环在辅助决策系统的 WEB 程序主要由下面两个部分组成：（1）数据显示部分（2）信息管理部分。数据显示部分主要负责将数据库中的数据转为面向用户的图形、曲线和表格等用户信息。信息管理部分主要负责网站的信息发布，用户权限管理等。

4.1.5 系统的总体架构

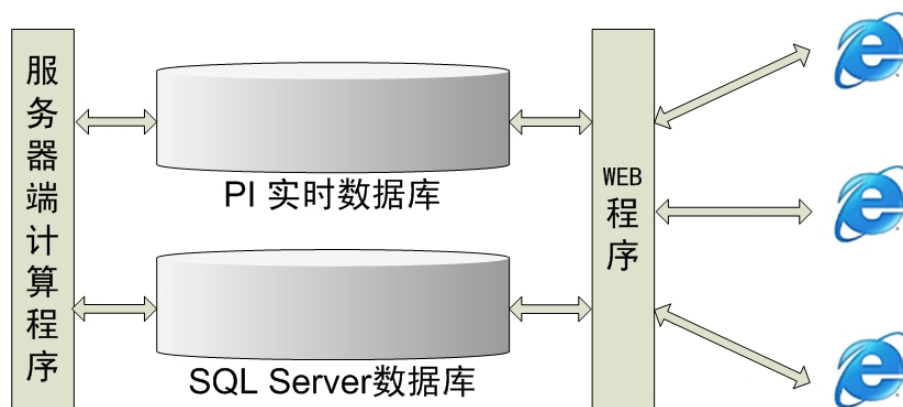


图 4.3 辅助决策系统总体架构

4.2 燃气和水蒸汽的热力性质计算

4.2.1 燃气热力性质计算方法

在半山发电厂联合循环机组辅助决策系统中对燃气轮机进行性能计算和分析时经常需要使用燃气热力性质的数据。燃气是天然气与空气燃烧后的产物，当温度小于 1500℃时，气体的离解现象可以忽略不计，因此它遵循理想气体的状态方程式，其比热和焓值均为温度的函数，与压强和比容无关。同时由于燃气是一种混合物，因此它的热力性质还与其组成成分的性质和比例有关。

传统燃气热力性质的计算方法是使用燃气热力性质表，如果将这种方法应用到电子计算机计算有很多缺点，首先它需要将整个表以一定的形式保存在计算机中供程序在需要时调用，输入数据的工作量很大。其次，燃气热力性质表只适用于特定的燃料，这就局限了它的使用范围。因此在电子计算机计算中，通常不使用这种方法，而是使用合适的公式表示燃气的热力性质之间的关系，使其程序化，从而方便编制计算机程序进行调用和查询。

燃气轮机的燃料主要由各种烃类组成，包含碳、氢、氧、氮、硫等元素，与空气混合燃烧后生成的燃气则包括二氧化碳、水蒸汽、二氧化硫、氮气以及过量的空气。因为半山发电厂联合循环机组辅助决策系统中的天然气不检测硫含量，因此通过实际测量换算得到的燃料的化学式为 $C_xH_yO_zN_u$ ，其燃料系数为 β ，它与理论空气量空气完全燃烧的反应方程式为

$$\beta \cdot C_xH_yO_zN_u + \left(x + \frac{y}{4} - \frac{z}{2}\right)(O_2 + dN_2) = \beta x CO_2 + \beta \frac{y}{2} H_2O + (1 - \beta) \left(x + \frac{y}{4} - \frac{z}{2}\right)(O_2 + dN_2) + \beta \left[\frac{u}{2} + d \left(x + \frac{y}{4} - \frac{z}{2}\right)\right] N_2 \quad (4.1)$$

式中：

$C_xH_yO_zN_u$ — 燃料的化学式

β — 燃料系数

d — 空气中氮气与氧气的体积比, 取 3.77382

这样在燃料各种烃类所占的体积百分比已知的情况下, 能够得到燃料各种元素的摩尔比从而得到燃气的成分及百分比, 最后通过理想气体的混合规则求出燃气的热力性质。

很多文献^{[33][34]}提供了适用于不同温度区间和不同燃料的燃气及其组分的热力性质的经验公式, 结合联合循环燃气轮机性能计算的需要, 半山发电厂联合循环机组辅助决策系统选择文献^[33]提供的公式, 该公式的适用温度范围为-50 到 1500℃。包括空气在内的大气氮和空气在内的气体热力性质的统一计算公式如下定压比热容:

$$c_p = \sum_{i=0}^8 a_{ci} \left(\frac{T}{1000} \right)^{i-1} \quad (4.2)$$

焓:

$$H = \sum_{i=0}^8 a_{hi} \left(\frac{T}{1000} \right)^i + b_h \ln \frac{T}{1000} \quad (4.3)$$

对数相对压力:

$$\ln \pi = \left[A_s + \sum_{i=0}^8 a_{si} \left(\frac{T}{1000} \right)^{i-1} + b_s \ln \frac{T}{1000} \right] / R_m \quad (4.4)$$

公式(4.2)~(4.4)中:

T — 燃气温度, 单位 K

c_p — 燃气定压比热容, 单位 kJ/kMol · K

H — 燃气焓, 单位 kJ/kMol

$\ln \pi$ — 燃气对数相对压力

R_m — 通用气体常数, 取 8.3144 kJ/kMol · K

文献^[33]给出了使用公式(4.2)到(4.4)计算空气、二氧化碳、水蒸汽和氮气各项热力性质的系数 (a_{ci} 、 a_{hi} 、 b_h 、 A_s 、 a_{si} 、 b_s) 的取值。计算燃气热力性质的系数由组成燃气的各种气体的体积百分比和系数决定。

$$E = \sum_{i=1}^4 r_i E_i \quad (4.5)$$

式中:

E — 组分性质计算公式中的参数, 如 a_{hi} 、 A_s 等

r — 组分的体积百分比

在燃气热力性质计算程序设计过程中, 把组成燃气的每种气体对应成为一个类, 燃气则对应燃气类, 每个类的初始化参数不一样。但是这些类有不少共同点,

如初始化操作、根据温度计算焓值等，因此可以把他们的这些共同点抽象出来建立一个基类，基类具有所有气体的共同属性和操作。从而减少了编程工作量，消除了冗余代码，又增强了程序的可读性。

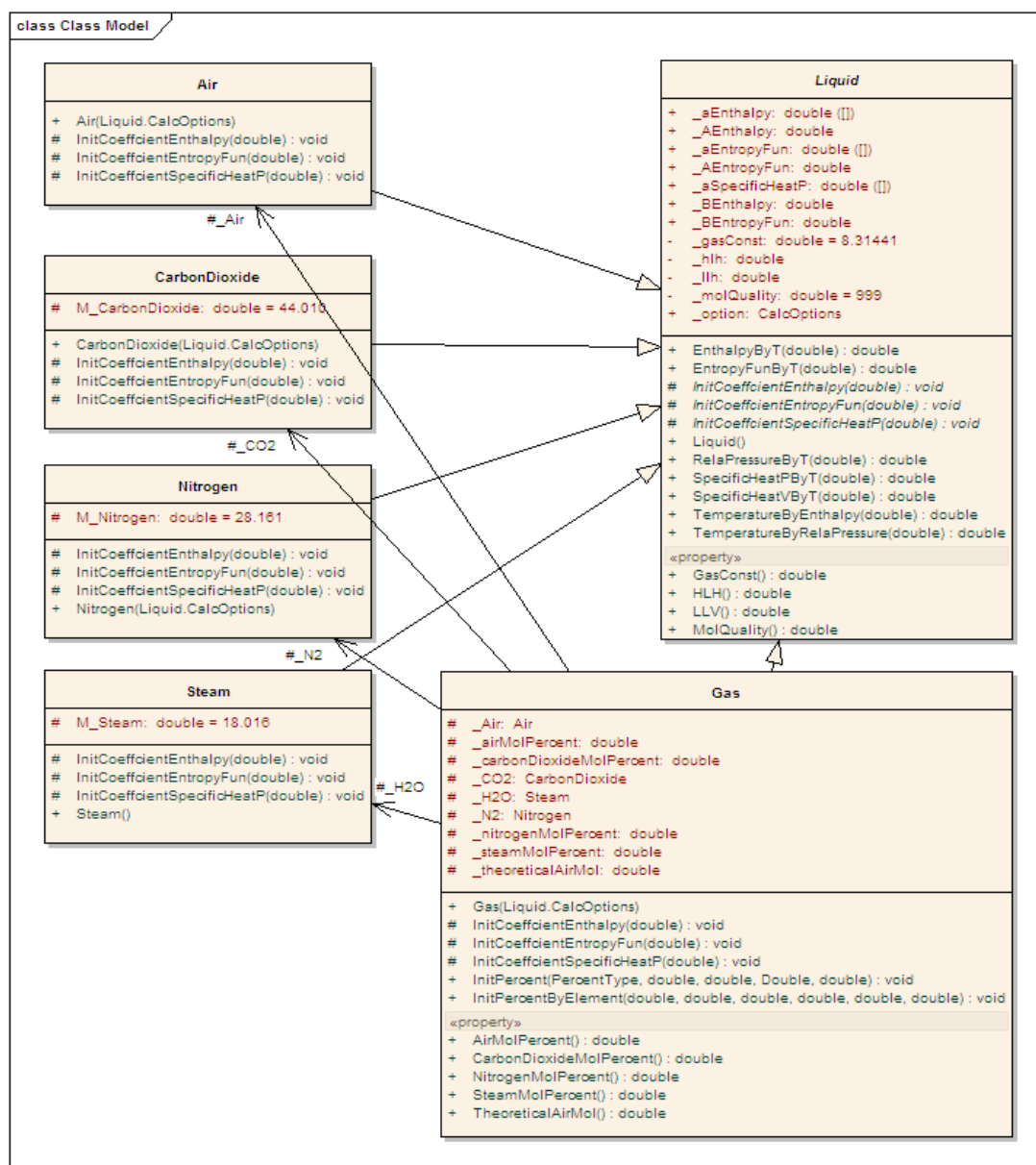


图 4.4 燃气热力性质计算命名空间结构

半山发电厂联合循环机组辅助决策系统中用于计算的类均位于 GasCalculation 命名空间中，流体类 (Liquid) 作为所有气体类的基类，提供了基本属性如摩尔质量、气体常数和基本方法如温度求取比热、焓值、对数相对压力等热力性质。由于它不代表某种具体气体，因此它是一个抽象类不能被实例化；各组成气体类 (Air、CarbonDioxide、Steam 和 Nitrogen) 派生自流体类，继承父类的属性和方法，并具体实现父类的初始化不同的参数抽象方法。燃气类 (Gas) 也派生自流体类并且增加相应属性如空气燃料比、燃料组成和方法如初始化组分

参数等。GasCalculation 命名空间中的类结构如图 4.4。

通过上述的面向对象的程序设计，简化了工作量并提高了程序的可读性，如通过温度计算燃气的热力性质，只需要通过如下语句便可以实现。

```
GasCalculation.Liquid gas = new GasCalculation.Gas(CalcOptions.USSR73);  
enthalpy = gas.EnthalpyByT(temperature);
```

通过该程序计算的结果与文献^[35]提供的燃气热力性质表查取的结果相比，在低温段两者吻合得很好，但是随着温度的升高，误差增加，这主要是由于经验公式及其参数都是基于燃气没有离解的情况下得到的。通过验证得到最大误差在 2% 以内，满足工程计算的要求。

4.2.2 水蒸汽热力性质计算方法

水和水蒸汽的热力性质在余热锅炉和蒸汽轮机性能计算及分析中使用非常广泛，它的状态的确定需要两个独立变化的热工参量（如温度和压力），由于水和水蒸汽不能被看作理想气体，因此它的热力性质计算非常复杂。传统的计算方法是使用水和水蒸汽热力性质图表，将这个用于计算机计算同样存在输入数据量巨大等缺点。

由于水和水蒸汽热力性质的复杂性，很难用单纯理论的方法确定它们的状态方程式，国际上著名的水蒸汽特性关系式，都是建立在实验的基础上，或是直接根据实验数据，或是根据国际水蒸汽会议采用的骨架表编制出来的。目前国内最为常用的水蒸汽特性关系式是公式化委员会 IFC 制定的水和水蒸汽热力性质。

在第六届国际水蒸汽性质会议上，成立了国际公式化委员会 IFC。IFC 在实验的基础上制定了国际公认的、工业用的水和水蒸汽热力性质公式。IFC 公式包括了骨架表的全部范围，根据公式计算出的数值都在骨架表规定的允许范围以内。现在最新的 IFC 公式是 IAPWS-IF97，包括饱和状态骨架表，比体积骨架表和比焓骨架表。该骨架表是根据当时对这些特性全部并被认为是可靠的测量结果编制而成的。

IAPWS-IF97 模型包含了用于 5 个分区的一系列公式，涵盖有效范围为： $273.15\text{K} \leq T \leq 1073.15\text{K}$ ， $P \leq 100\text{MPa}$ 和 $1073.15\text{K} \leq T \leq 2273.15\text{K}$ ， $P \leq 10\text{MPa}$ 。根据国际水与水蒸汽性质协会提供的数据以及利用最新编制的通用计算软件的核算结果，以国际公认的骨架表数据为准，IAPWS-IF97 的精度要比 IFC-67 高出一个数量级并且平均计算速度快 5 倍左右。

在半山发电厂联合循环机组辅助决策系统中将直接使用已根据 IAPWS-IF97 公式编制好的动态链接库，该动态链接库提供了水和水蒸汽热力性质计算函数的入口，在 C# 中可以使用 3.2.3 节所述的 P/Invoke 技术直接调用它，但为了方便在程序中的调用，半山发电厂联合循环机组辅助决策系统将其包装成了一个静态类。

4.3 维修间隔计算的实现

维修间隔计算的实现同时需要 PI 数据库和普通关系数据库的支持，它从 PI 数据库中取出实时/历史数据进行分析，并将统计完成的记录保存到 SQL Server 数据库中。维修间隔计算使用到的 PI 测点有火焰信号、总点火时间、总启动次数、总跳闸次数和发电机有功功率，其中火焰信号用于判断机组的启停状况、点火时间用于计算机组的运行时间和停机时间，启动次数用于辅助判断记住的启停，跳闸次数用于判断机组是跳闸，发电机有功功率则用于判断机组的最大负荷和跳闸负荷。SQL Server 数据库中有五个相关的表用于存储初始化数据和历次统计结果，他们分别是用于存储初始化数据、每次启停数据、燃烧系统维修周期统计数据、热通道维修周期统计数据和转子维修周期统计数据。

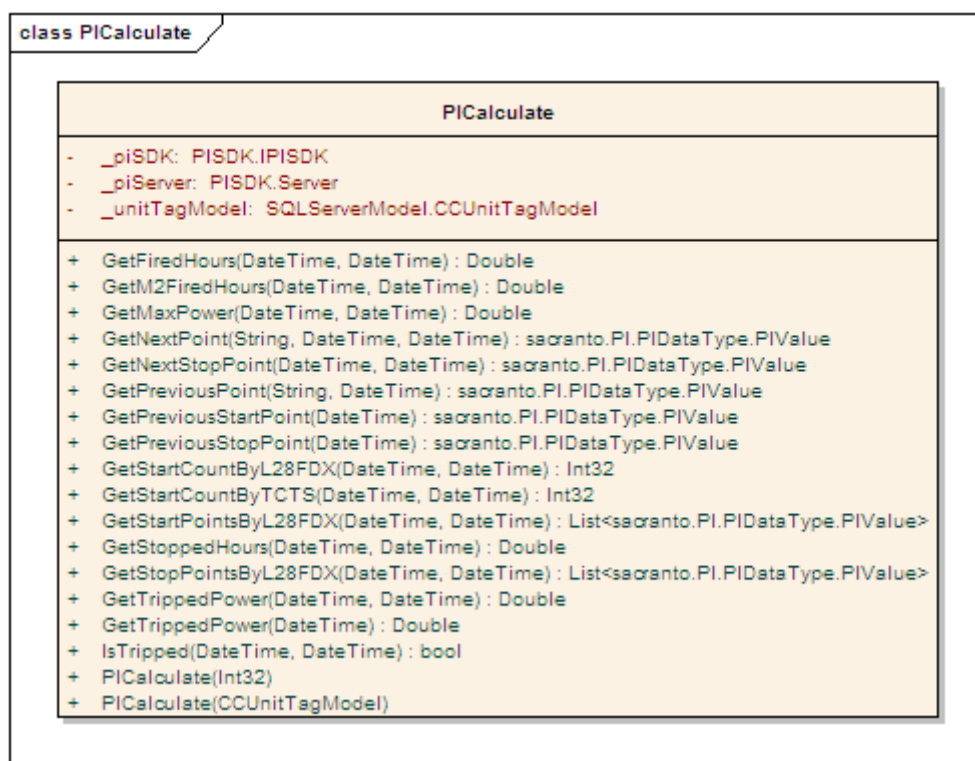


图 4.5 PICalculate 类视图

维修间隔的数据一部份保存在 PI 实时/历史数据库中，还有一部份初始数据是人工记录的，计算时首先对人工记录的数据进行分析统计，一次性在计算初始化过程中将它们保存到记录每次启停数据统计和各部件维修周期统计的表中。初始化完成以后，系统开始根据 PI 数据库中的记录自动维护和统计维修间隔数据。无论是根据人工记录还是 PI 数据库中的测点记录进行维修间隔计算，其基础都是得到机组的最大负荷、停机时间、是否跳闸等参数，在半山发电厂联合循环机组辅助决策系统中使用了 PICalculate 类来专门计算上述参数，其中使用 PI-SDK 来获取 PI 系统中的数据。

PICalculate 类的构造函数用于得到需要计算的机组。

GetStartCountsByL28FDX、GetStopPointsByL28FDX 方法用于得到给定时间的起始时间和结束时间用点火信号判断的启动次数或者停机次数，两者均返回 sacramento.PI.PIDataType.PIValue 类型的集合，该类型是一个数值-时间对，类似但不同于 PI-SDK 中的 PIValue 对象，它自动把 digital 类型的值表示为数值，而不是 Com 对象。

GetPreviousStartPoint 用于返回给定时间对应的机组启动点。这样在使用 GetStopPointsByL28FDX 函数得到停机点的情况下再利用 GetPreviousStartPoint 函数就能够得到每次启动停机对应的时间点，从而可以进一步利用 GetFiredHours 函数得到每次启停的运行小时、利用 GetMaxPower 函数得到每次启停过程中的最大负荷。

GetPreviousStopPoint 用于返回给定时间对应的机组上一次停机点，同上，在使用 GetStartCountsByL28FDX 函数得到启动点的情况下再利用 GetPreviousStopPoint 函数就能够得到每次启动停机对应的上一次停机的时间点，从而可以进一步可以利用 GetStoppedHours 函数得到每次的停机小时数。

IsTripped 使用总跳闸次数测点判断在给定的时间内是否存在跳闸，由于只要测点数据变化就会被记录为跳闸，因此使用该函数时给定的时间通常在一次启停段内。GetTrippedPower 用于在总跳闸次数增加的时间点周围取三个功率点并选择其中最大值作为跳闸功率。

在高一级的层次上，半山发电厂联合循环机组辅助决策系统维修间隔计算模块使用 PICALculate 类计算上次统计时间到当前的启动点和停机点，如果该段时间内没有启动和停机则不进行任何计算直接返回，如果存在启动或者停机则进行相应的启停计算。其流程如图 4.6。

在自动维护时刻到程序开始运行时刻的周期或者计算周期设置比较长时，可能在一个计算周期内存在多次启动和停机的状况。为了避免停机计算过程和启动

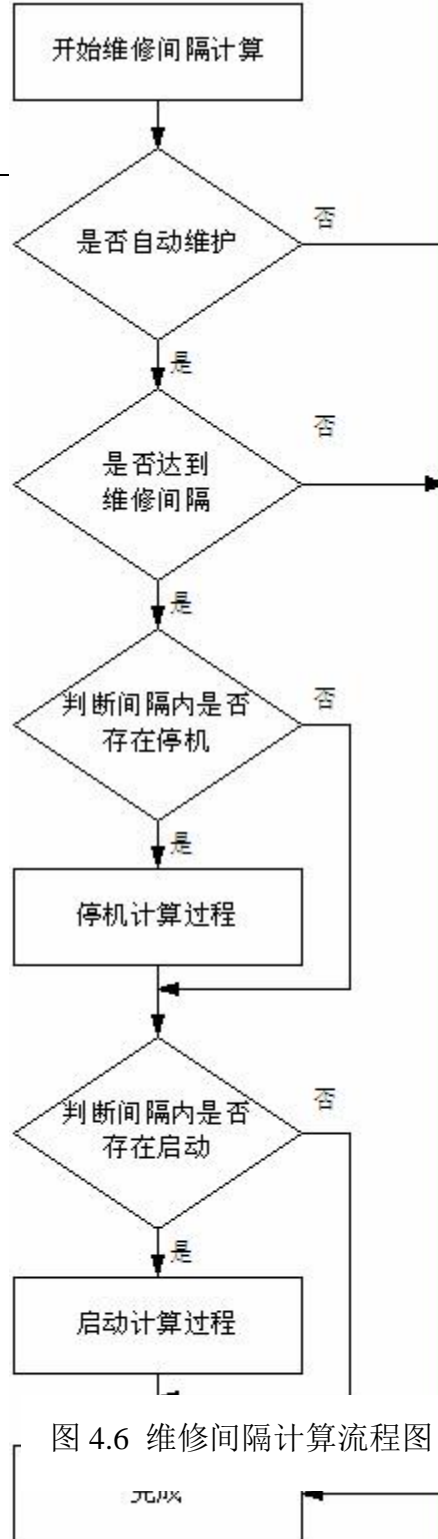


图 4.6 维修间隔计算流程图

计算过程中进行重复计算, 如果一个周期内存在多个启停点, 半山发电厂联合循环机组辅助决策系统维修间隔将在启动过程中统计每次启停数据, 停机计算过程只处理最早的一次停机, 并且判断该次停机对应的启动点是否也在这个计算周期内, 如果对应的启动点也在这个周期内, 停机计算过程将不进行计算直接返回, 因为它在启动计算过程中已进行。

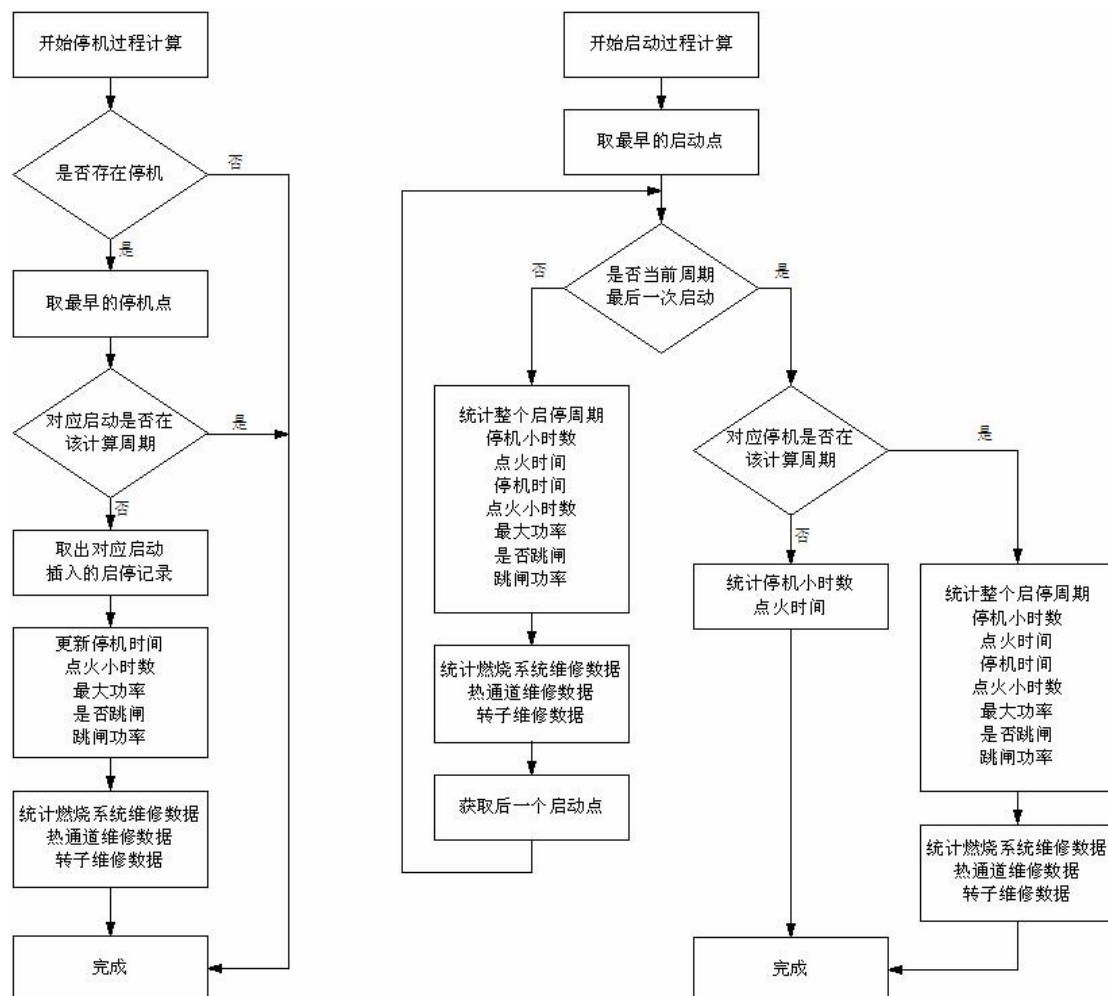


图 4.7 停机过程计算和启动过程计算流程图

停机过程计算需要进行 4 类统计并将他们存储到响应的数据库, 分别是启停数据统计、燃烧系统维修周期数据统计、热通道维修周期数据统计和转子维修周期统计。计算首先从启停数据表中得到停机对应的在启动时插入的当前记录, 然后将停机时间、点火小时数、最大功率、是否跳闸、以及跳闸功率更新, 这样得到了一天完整的启停记录, 最后再根据这条启停记录, 利用部件维修间隔计算模型分别对各个部件进行并更新至数据库。

启动过程计算相比停机过程计算要复杂一些, 因为它要统计计算周期内的所

有启停过程。对应停机点不在该计算周期的启动点只需新建一条启停记录，统计停机小时数和点火时间并将该记录插入到数据中。对应停机点也在这个计算周期的启动点要完整统计该启停过程，即新建一条启停记录，不仅统计停机小时数和点火时间，还要实现上述停机过程计算的步骤。

停机过程计算和启动过程计算的步骤如图 4.7。

4.4 在线性能计算的实现

半山发电厂联合循环机组辅助决策系统在线性能计算模块的流程与原理部分所述基本相同，按照压气机—燃烧室—燃气透平—燃气轮机组—余热锅炉—蒸汽轮机—联合循环机组的顺序依次计算。每个部件的具体计算方法和步骤在原理部分已经详细给出，在此不再赘述。

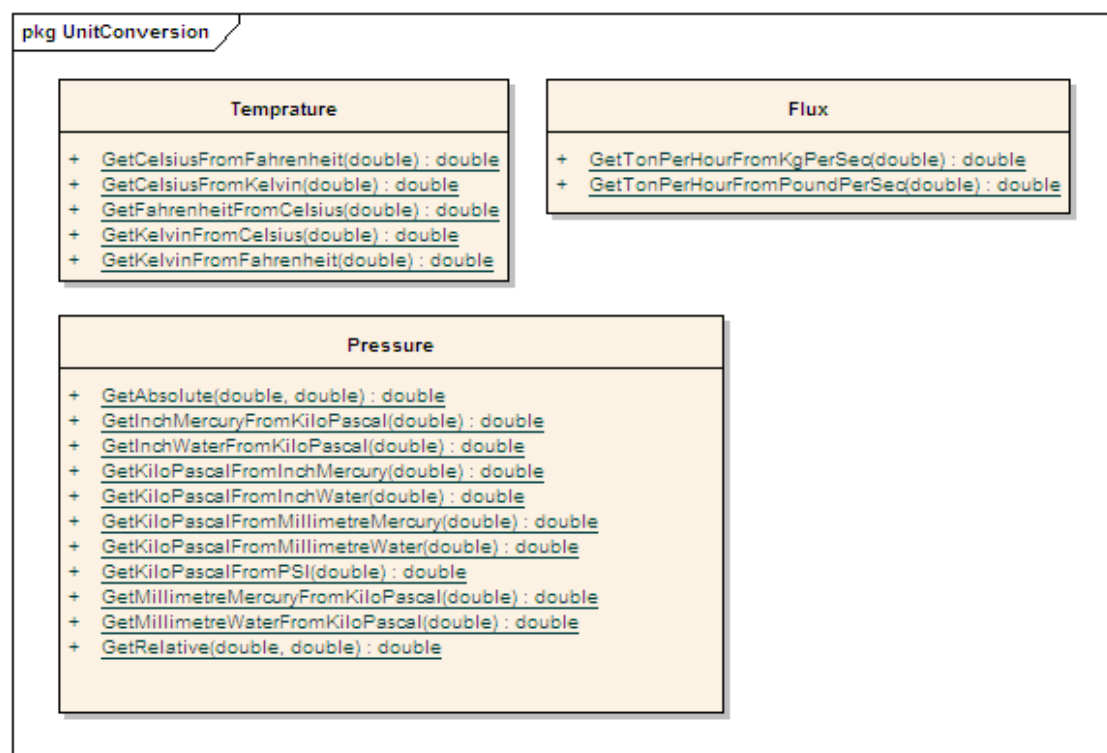


图 4.8 UnitConversion 命名空间图

在线性能计算涉及到 PI 数据库中的许多测点，从其中取出实时数据加以分析和计算。由于现场使用的数据采集传感器不同，导致 PI 数据库的测点单位不尽一致，涉及测点使用到温度单位有摄氏度和华氏度，压强单位有磅/平方英寸、兆帕和千帕等，这些单位与计算过程中预设的单位可能不一样，因此需要对它们进行转换。为了统一和方便进行单位换算，半山发电厂联合循环机组辅助决策系统把处理单位转换的类均放到使用 UnitConversion 命名空间中。

在线性能计算中使用到的数据还有一个特点，即它们通常是某种工质的参数，

如压气机出口的空气温度、压强和流量，高压缸进口蒸汽温度、压强和流量等，在实际程序设计中把它们均抽象成为 **Workmaterial** 类，使用中只需要指定它的温度和压强还有工质类型，它便自动计算工质对应的焓值。

4.5 辅助决策系统计算结果

通过对辅助决策系统的计算结果进行分析，以验证程序和模型相符合。

对于维修间隔计算，本文截取了半山发电厂一台机组某时间段的启停记录，如下表，表中数据为计算程序根据 **PI** 数据库中的历史数据进行统计，程序统计结果与电厂运行日志记录符合。

表 4.1 某机组启停记录列表

启动时间	停机时间	最大功率	是否跳闸	跳闸功率
2007-01-12 07:01	2007-01-12 19:28	391.9295959	0	
2007-01-13 07:55	2007-01-13 17:08	393.6843872	0	
2007-01-14 07:49	2007-01-14 18:00	404.5180969	0	
2007-01-15 07:48	2007-01-15 14:33	378.616394	0	
2007-01-28 07:11	2007-01-28 09:09	56.12182999	1	56.12183
2007-01-28 11:35	2007-01-28 21:40	382.1640015	0	
2007-01-29 08:45	2007-01-29 19:47	406.2347107	0	
2007-01-30 08:41	2007-01-30 16:47	339.2486877	0	
2007-01-31 08:46	2007-01-31 17:04	393.8750916	0	
2007-02-01 06:26	2007-02-01 17:02	396.0114136	0	
2007-02-02 07:08	2007-02-02 18:21	363.6626892	0	
2007-02-03 07:00	2007-02-03 18:35	383.8806152	0	
2007-02-04 07:24	2007-02-05 21:53	391.0903931	0	
2007-02-06 08:41	2007-02-06 14:44	311.2106018	1	280.8838
2007-02-06 16:35	2007-02-06 21:30	352.7526855	0	
2007-02-07 09:18	2007-02-07 20:58	365.1885986	0	
2007-02-08 07:58	2007-02-08 23:01	385.9024048	0	
2007-02-09 07:32	2007-02-10 05:24	393.4173889	0	
2007-03-27 09:25	2007-03-28 20:04	368.6980896	0	
2007-03-30 06:00	2007-03-30 21:19	363.357605	0	
2007-03-31 07:13	2007-03-31 16:22	293.47229	0	
2007-04-18 06:20	2007-04-19 18:41	384.1094971	0	
2007-04-20 07:06	2007-04-21 18:25	359.8479919	0	
2007-04-22 07:26	2007-04-22 07:49	0	0	
2007-04-22 10:05	2007-04-23 18:28	385.2539063	0	

根据上述结果统计考虑各种因素影响的燃烧系统、热通道和转子的运行时间和启动次数如表 4.2。

表 4.2 某机组维修间隔计算统计结果

设备	程序计算	程序计算	人工计算	人工计算
----	------	------	------	------

	运行时间	启动次数	运行时间	启动次数
燃烧系统	373.1	27	373.1	27
热通道	373.1	34	373.1	34
转子	373.1	30	373.1	30

对于在线性能计算，本文根据制造厂提供的热力系统图^[13]中的参数校核设备的等熵效率计算，由于热力系统图中燃气轮机部分参数没有提供，因此只核算余热锅炉和蒸汽轮机部分的等熵效率；机组的功率和热耗率修正计算参照机组性能试验资料校核。结果如表 4.3.

表 4.3 某机组在线性能计算部分结果

参数	程序计算结果	性能试验结果
锅炉反平衡效率	0.6108	0.6108
锅炉正平衡效率	0.6088	0.6095
蒸汽轮机高压缸效率	0.8516	0.8484
蒸汽轮机中压缸效率	0.9038	0.9126
蒸汽轮机低压缸效率	0.7162	0.7162
联合循环功率	372875.8	372875.8
功率_大气温度修正	1.0261	1.0262
功率_大气压力修正	0.997	0.997
功率_大气相对湿度修正	0.999998	0.99998
功率_频率修正	0.9996	0.9996
功率_低位发热量修正	0.9579	0.9997
功率_排气压力修正	-4686.09	-4687.3
功率_燃气加热器进口燃料温度修正	8.3	8.3
功率_功率因素修正	880.68	880.78
功率_点火总小时数修正	1.0094	10.0094
修正后联合循环功率（未考虑老化）	384867	384462
修正后联合循环功率（考虑老化）	388495	388840
联合循环热耗率	6342.24	6352.96
热耗率_大气温度修正	1.0258	1.0258
热耗率_大气压力修正	0.9969	0.9969
热耗率_大气相对湿度修正	0.999937	0.999926
热耗率_频率修正	0.9997	0.9996
热耗率_低位发热量修正	0.9993	0.9998
热耗率_点火总小时数修正	1.0032	1.0032
修正后联合循环热耗率（未考虑老化）	6276.93	6286.47
修正后联合循环热耗率（考虑老化）	6257.01	6266.51

从表 4.2 和表 4.3 可以看出维修间隔和在线性能计算结果与人工计算和性能试验数据基本符合，从而证明系统程序的计算结果是正确的。

此外，为了检验联合循环机组在线性能计算模型和程序的正确性，本文选取

了辅助决策系统在半山发电厂试运行期间的部分功率数据，功率分布于 290MW 到 390MW 之间，如表 4.4。通过对热力计算得到的燃气轮机功率和蒸汽轮机功率之和与机组实测功率进行比较，两者的相对误差不超过 4%，说明模型和程序是可靠的。

表 4.4 计算功率与实测功率的比较

计算燃气轮机功率	计算蒸汽轮机功率	计算联合循环机组功率	实测功率	相对误差 (%)
182355.9	111060.6	293416.5	299957.2	2.180544424
211123	106840.8	317963.8	321777.3	1.185136428
221721.7	115567.4	337289.1	350502	3.769707448
215686.3	125670	341356.3	347908	1.883170263
233635.8	124610	358245.8	360229.5	0.550676721
246535.8	127028.3	373564.1	385711.6	3.149373781

4.6 辅助决策系统截图

4.6.1 服务器端计算程序

服务器端计算程序由主窗口和选项窗口组成。主窗口左边让用户选择进行计算的机组和开启或者停止计算，右边显示机组的上次性能计算时间。见图 4.9。

选项窗口用于服务器端计算程序初始化时对应用程序进行配置，通过它可以设置计算间隔，修改数据库连接字符串和切换维修间隔计算模式等。见图 4.10。



图 4.9 服务器端计算程序主界面

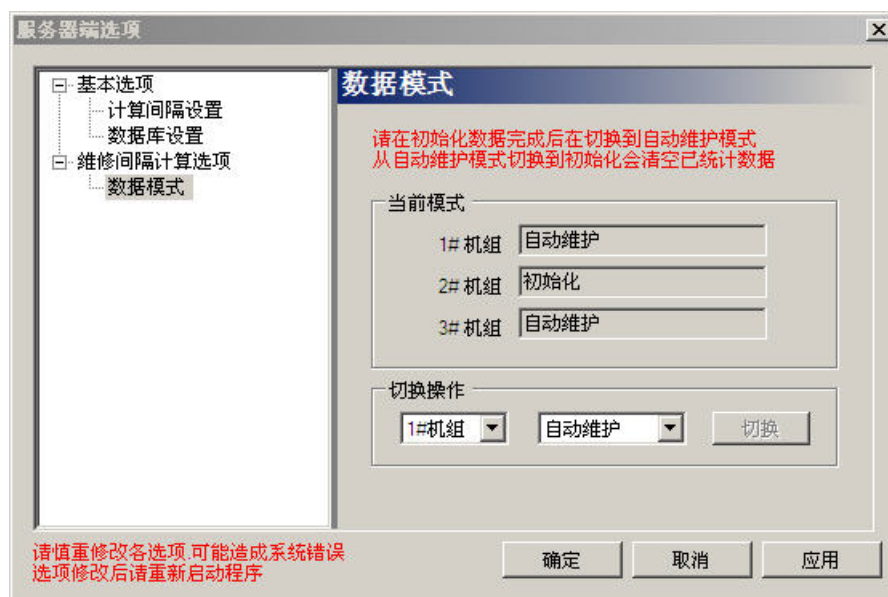


图 4.10 服务器端计算程序选项界面

4.6.2 WEB 应用程序

WEB 应用程序由信息浏览页面和管理员页面组成。信息浏览页面可以匿名浏览，它包括三个区域，上方的导航栏用于选择机组，左边的菜单栏用于选择进入相应功能的界面，内容显示于右方，根据菜单中的选项进行变化。整个页面使用 Ajax 技术实现无刷新导航，提高用户体验。见图 4.11。

管理员页面需要分配的用户名和密码才能进入，并且不同的管理员对应于不同的权限，例如维修间隔计算管理员只能进入维修间隔菜单及其子菜单中的页面，对维修间隔计算进行维修确认，设置初始化数据等操作。见图 4.12。

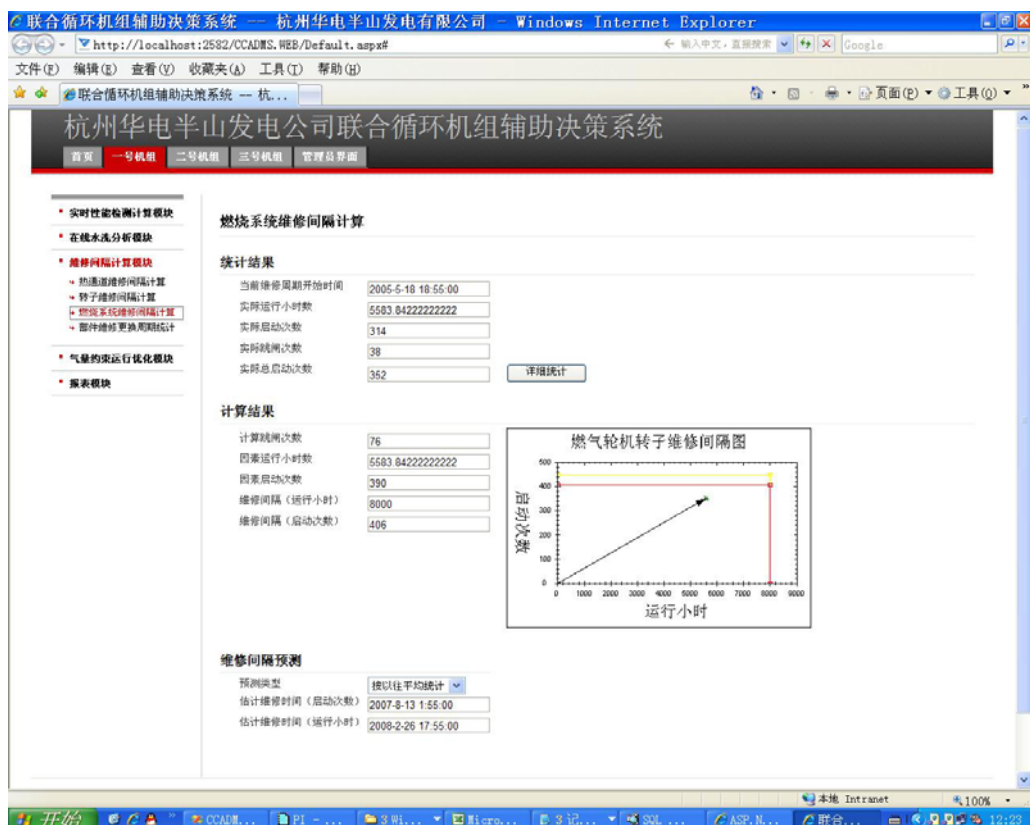


图 4.11 WEB 应用程序信息浏览界面

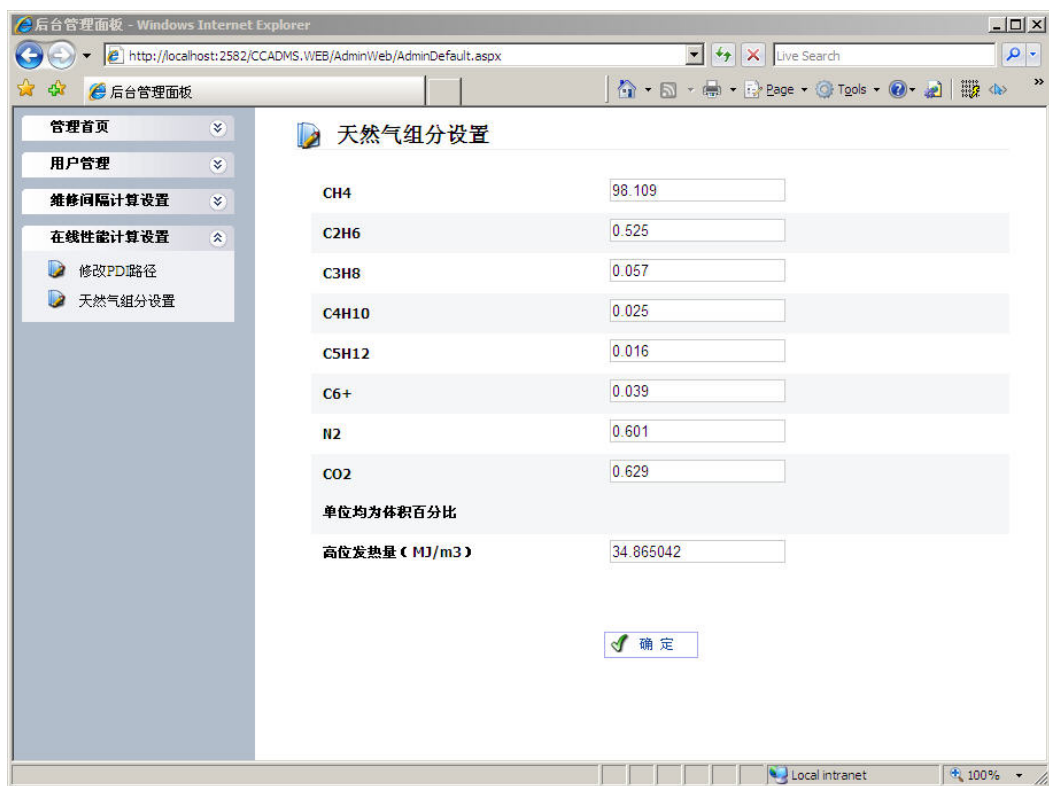


图 4.12 WEB 应用程序管理员界面

第五章 结论与展望

本文结合了燃气蒸汽联合循环机组的基本原理和制造厂提供的设计与性能试验资料对单轴重型联合循环机组建立维修间隔计算和在线性能计算方法和模型,并在此模型上使用相关的计算机技术开发出基于 B/S 架构的发电厂联合循环机组辅助决策系统,帮助电厂运行人员提高维修和运行水平。论文的研究内容和研究成果总结如下:

1. 总结了针对 GE 公司生产的重型燃气轮机的维修间隔计算和设备更换模型,结合 PI 数据库中的数据实时计算各台联合循环机组考虑外因影响后的因素启动次数和因素运行时间,并利用数据列表和图形等方式直观的给出机组最大运行周期,从而指导电厂维修人员制定合适的维修计划,使停机时间最少,在保证设备可靠性的基础上得到设备的最大可用率。

2. 研究了单轴联合循环机组的在线性能计算方法,给出燃气轮机、余热锅炉、蒸汽轮机和联合循环机组整体的效率和热耗率等电厂运行关键技术指标,用以提高机组运行水平。此外还通过热力计算得到不能直接测量的燃气轮机和蒸汽轮机的出力,辅助运行人员判断燃气轮机和蒸汽轮机的性能。

3. 联合循环机组的功率和效率受外部因素影响比较大,本文对受到大气参数、燃料低位发热量和发电机频率等外部因素影响的联合循环的热效率和热耗率根据制造厂提供的性能曲线和拟合公式进行修正,用于比较不同时间和不同环境下的运行水平,并且为机组以后进行负荷分配优化提供指导数据。

4. 讨论了联合循环在线性能计算中使用的燃气热力性质的计算方法,并利用面向对象编程思想实现燃气热力性质计算模块。

5. 使用微软公司最新的.NET 程序设计框架完整实现半山发电厂联合循环机组辅助决策系统。.NET 框架的核心为公共语言运行时和基础类库,它提供了许多优于传统编程模型的特性,并在 CLR 的基础上提供了适用于开发不同类型程序的程序设计语言、开发 WEB 应用程序和 WEB 服务的 ASP.NET 以及用于连接普通关系数据库的 ADO.NET。

6. 讨论和实现利用.NET 框架通过 PI 数据库提供的应用程序接口或者软件开发工具对 PI 实时/历史数据库进行开发的方法,其中涉及到 PI-API 函数库、PI-SDK 层次数据模型、P/Invoke 技术及其数据封送处理和.NET 对 COM 组件的调用方法等技术。

7. 在半山发电厂联合循环机组辅助决策系统的实现过程中应用了对象关系映射(ORM)和异步 JavaScript 和 XML(Ajax)等最新的软件设计思想和技术提高程序的可扩展性和用户体验。通过 ORM 技术既可利用面向对象程序设计语言的简单易用性,又可利用关系数据库的数据操作优势。通过 Ajax 技术能够在不更新整个页面的情况下进行数据传输和维护,并且避免发送重复信息,使得

WEB 应用程序更为迅捷地响应用户操作并且提高用户体验。

半山发电厂联合循环机组辅助决策系统的计算结果与电厂的热力试验和人工计算统计结果的误差一般在 2% 以内，最大的误差是热力计算出力和实测功率差值，两者最大相对误差为 4%。

半山发电厂联合循环机组辅助决策系统主要有两点需要改进，第一是性能计算中燃气轮机进口温度的计算方法，本文采用了假设燃烧室效率通过燃烧室热平衡求取的方法，它的主要缺陷在于不能实际计算燃烧室效率这一比较重要的热力指标。第二是使用 Ajax 技术实现 WEB 应用程序无刷新导航时利用 ViewState 保存页面状态从而增加了服务器和浏览器之间的数据传输量。这两点有待在以后的工作进行优化。

参考文献

- [1] 焦树建. 燃气—蒸汽联合循环[M]. 北京: 机械工业出版社, 2000.
- [2] 杨 洋. 联合循环机组运行技术的研究[D]. 华北电力大学, 2006.
- [3] 陈坚红. 联合循环机组运行计划和负荷分配: 建模、启发式遗传算法求解和数据处理[D]. 浙江大学, 2004.
- [4] GE Energy. Heavy-Duty Gas Turbine Operating and Maintenance Considerations[Z]. 2004.
- [5] 曹 宏. 联合循环电站性能计算软件开发[D]. 重庆大学, 2002, 5.
- [6] 白慧峰, 徐 越, 危师让等. 单轴燃气—蒸汽联合循环机组热力性能模拟计算软件的开发. 燃气轮机技术[J], 2004, 3: 40-43.
- [7] 姜焕农. F型单轴联合循环气耗率估算须考虑的因素[J]. 燃气轮机技术, 2003 12: 31-33.
- [8] 孙福珠. 单轴联合循环发电机组性能分析系统[J]. 国际电力, 2003, 4: 16-17.
- [9] 王德慧, 李 政, 麻林巍等. 大型燃气轮机冷却空气量分配及透平膨胀功计算方法研究[J]. 中国电机工程学报, 2004, 1: 180-185.
- [10] 沈炳正. 燃气轮机装置[M]. 北京: 机械工业出版社, 1981.11.
- [11] J.H.Horlock, D.T.Watson, T.V.Jones. Limitations on Gas Turbine Performance Imposed by Large Turbine Cooling Flows[J]. Journal of Engineering for Gas Turbines and Power, 2001, 7
- [12] 郑莆燕, 姚秀平, 齐 进等. 确定运行中的燃气轮机初温的方法探讨[J]. 燃气轮机技术. 2005, 3: 57-59.
- [13] GE Power Systems. Field Performance Testing Procedure[Z]. 1997.
- [14] Don Box, 张晓坤译. .NET本质论[M]. 北京: 中国电力出版社, 2004.
- [15] Jeffrey Richter, 李建忠译. Microsoft.NET框架程序设计[M]. 北京: 清华大学出版社, 2003.
- [16] Robinson Simon, Nagel Christian, 李敏波译. C#高级编程[M]. 北京: 清华大学出版社, 2005.
- [17] Stefan Schackow. Professional ASP.NET 2.0 Security, Membership, and Role Management[M]. Indianapolis: Wiley Publishing, 2006.
- [18] 苟建兵, 侯子良. 电厂厂级实时监控信息系统建设探讨[J]. 电力设备. 2001, 2: 20-23.
- [19] OSI Software. PI ProcessBook User Guide[Z]. 2002.

- [20] OSI Software. PI ProcessBook Programmer Reference[Z]. 2003.
- [21] OSI Software. Advanced PI Client ActiveX Scripting[Z]. 2006.
- [22] OSI Software. PI-ActiveView Developer Guide[Z]. 2002.
- [23] OSI SOFTWARE. Practical Examples of PI-ActiveView, COM, UDA, the Module Database, and Sigmafine[Z]. 2000.
- [24] MSDN. Microsoft. Activating ActiveX Controls[EB/OL]. http://msdn.microsoft.com/library/default.asp?url=/workshop/author/dhtml/overview/activating_activex.asp.
- [25] OSI Software. PI-API Programmer's Help[Z]. 2003.
- [26] Jason Clark. Calling Win32 DLLs in C# with P/Invoke[EB/OL]. <http://msdn.microsoft.com/msdnmag/issues/03/07/NET/>. 2003, 7.
- [27] OSI Software. PI-SDK Programmer's Help[Z]. 2003.
- [28] Mike Gunderloy. Calling COM Components from .NET Clients[EB/OL]. <http://msdn2.microsoft.com/en-us/library/ms973800.aspx>. 2001, 11.
- [29] Sam Gentile. 添加引用的背后: 有人看见桥了吗? [EB/OL]. http://www.microsoft.com/china/msdn/archives/library/dndotnet/html/bridge.asp#br_topic6. 2003, 10.
- [30] Gregory Leake. Microsoft .NET Pet Shop 4: Migrating an ASP.NET 1.1 Application to 2.0[EB/OL]. <http://msdn2.microsoft.com/en-us/library/aa479070.aspx>. 2006, 2.
- [31] Dave Crane, Eric Pascrello, Darren James等[M]. Ajax实战. 北京: 人民邮电出版社, 2006.
- [32] 无刷新页面跳转[EB/OL]. <http://www.cnblogs.com/hjf1223/archive/2006/06/20/430475.html>. 2006, 6.
- [33] 刘志刚, 刘咸定, 赵冠春等. 工程热物理性质计算程序的编制及应用[M]. 北京: 科学出版社, 1992.
- [34] 张士铮. 燃气热力性质的数学公式表示法[J]. 工程热物理学报, 1980, 2: 9-16.
- [35] 严家騄, 杨玉顺, 刘 明. 烃类燃料的燃气热力性质表[M]. 北京: 科学出版社, 1989.

研究生期间的科研工作和发表论文

科研工作：

2005 年 12 月 — 2007 年 3 月 杭州华电半山发电有限公司联合循环机组辅助决策系统项目。主要负责联合循环机组维修间隔计算和在线性能计算的数学模型建立，程序架构设计和完整实现。

发表论文：

刘 昱，陈坚红，盛德仁等．基于 Clips 的可视化专家系统开发[J]．计算机应用与软件，已录用．

致谢

毕业论文从开题、项目选择、资料收集到最终定稿，得到了许多老师和同学的无私帮助。在此我对他们表示衷心的感谢。

在浙江大学的两年研究生生活和学习中，我的导师盛德仁教授、陈坚红副教授以他们丰富渊博的知识给了我极大的帮助，使我在科研学习中少走了很多弯路，同时他们给我提供便利的条件，让我能够顺利的完成科研工作。他们一丝不苟的工作态度、求实创新的治学作风更给我留下了深刻的印象，使我受益匪浅。

同课题组的李蔚老师、任浩仁教授也给了我非常大的帮助和鼓励，向他们请教和与他们进行讨论的过程中，我得到了很多关于问题思考和研究方法的启发。同时还要感谢半山发电有限公司的工作人员，他们提供的资料、数据和现场调试环境给我的科研提供了支持。此外我还要感谢实验室的同窗们在学习生活中给予我的帮助。

最后，我要感谢我的父母和家人，正是他们在背后的理解、支持和鼓励，是我克服生活和学习过程中各种困难的动力源泉。