

分类号 \_\_\_\_\_  
UDC \_\_\_\_\_

密 级 \_\_\_\_\_  
编 号 10490

武 汉 工 程 大 学  
工程硕士研究生硕士学位论文

**题目：聚丙烯装置的集散控制系统和  
风险可操作性分析系统设计与优化**

研 究 生 姓 名：王瑜

指导教师姓名、职称：宣爱国 副教授

李晓林 副教授

工 程 领 域 名 称：化学工程

研 究 方 向：

2009 年 5 月



## 独 创 性 声 明

本人声明所呈交的学位论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除文中已经标明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对本文的研究做出贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：

年 月 日

## 学位论文版权使用授权书

本学位论文作者完全了解我校有关保留、使用学位论文的规定，即：我校有权保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅。本人授权武汉工程大学研究生处可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

保 密 ☐ ， 在 \_\_\_\_\_ 年解密后适用本授权书。

本论文属于

不保密 ☐ 。

（请在以上方框内打“√”）

学位论文作者签名：

指导教师签名：1、

年 月 日

2、

年 月 日



**A Thesis Submitted in Fulfillment of the Requirements  
for the Degree of Master of Engineering**

**Design and optimize of the DCS and HAZOP system for  
the production process of polypropylene**

**Major : Chemical engineering**

**Candidate : Wang Yu**

**Supervisor : 1、Xuan AiGuo  
2、Li XiaoLin**

**Wuhan Institute of Technology  
Wuhan, Hubei 430074, P. R. China  
May 2009**



摘要.....	11
关键词: .....	12
ENGLISH ABSTRACT: .....	13
第 1 章 引言 .....	15
第 2 章 文献综述 .....	16
2.1 聚丙烯工业、工艺的发展状况 .....	16
2.2 聚丙烯生产特点 .....	18
2.3 聚丙烯装置概况 .....	19
2.4 现代控制系统与在线监测技术的发展状况 .....	22
2.5 计算机软件发展的趋势 .....	23
2.6 本课题的研究内容与创新点 .....	24
第 3 章 DCS 系统的组成和特点.....	25
3.1 DCS 系统的组成和特点.....	25
3.2 HAZOP 分析法.....	26
第 4 章 系统分析与设计 .....	29
4.1 系统需求分析 .....	29
4.2 DSC 系统的架构和通信方式.....	30
4.3 系统整体设计.....	32
4.3.1 为何要使用面向对象的分析方法分析 DCS 系统.....	33

4.3.2 DCS 系统功能模块分析 .....	33
4.3.3 开发环境的确定 .....	37
4.4 监听器的设计 .....	38
4.5 数据处理模块 .....	40
4.5.1 数据处理模块中的内存处理算法 .....	40
4.6 测量点对象的设计 .....	41
4.7 数据结构设计 .....	42
4.8 设备对象的设计 .....	43
4.9 系统的内存管理 .....	45
4.10 本章小结 .....	47
第 5 章 聚丙烯装置的 HAZOP 模块的设计 .....	47
5.1 控制点对象的设计 .....	47
5.2 数据的收集和处理 .....	49
5.2.1 数据的收集 .....	49
5.2.2 数据的持久化 .....	50
5.2.3 Hibernate 持久层的实现 .....	52
5.2.4 连接池的使用 .....	54
5.3 HAZOP 模块的设计 .....	55
5.3.1 HAZOP 模块与 DCS 系统的整合 .....	56
5.4 用户控制界面设计 .....	57



5.4.1 SVG.....	57
5.4.2 Ajax.....	61
5.4.3 DWR.....	62
5.5 配置文件设计 .....	65
5.6 测试平台设计 .....	67
5.6.1 测试平台设计出发点 .....	68
5.6.2 压力测试和异常测试 .....	68
5.6.3 测试平台的组成 .....	69
5.7 多线程协同工作 .....	72
5.8 本章小结 .....	74
第六章 结论.....	74
6.1 研究结果 .....	74
6.2 测试情况 .....	75
6.3 研究解决的问题 .....	77
6.4 研究的展望 .....	77
6.5 结论 .....	78
参考文献.....	79
致谢.....	82



## 摘要

随着科学技术的进步和新资源的不断开发,我国的石油化工得到飞速发展。石油化工由于生产过程具有高温、高压、易燃、易爆、腐蚀、剧毒等特点而成为事故多发行业。

武石化 10 万吨/年聚丙烯生产线于 98 年 6 月投产,至今已经运行了 10 年。为了降低生产中出现的风险,本项目计划将 HAZOP 分析法引入到生产系统中,一方面改进现有 DCS 系统,用新的软件技术手段优化 DCS 系统的展示及数据处理;一方面提高该系统对可能出现的风险具有一定的感知能力。

本文介绍了聚丙烯工艺,生产技术现状和发展方向。阐述了 HAZOP 分析法的定义以及如何降低化工生产中的风险。从软件设计的角度上,遵循面向对象的分析法,对 DCS 系统各个部分进行了分析和抽象。DCS 系统根据功能划分,可以分为:监听器,测量点对象,测量点组,控制点对象等。将整个系统模块化后,使得 HAZOP 分析法模块可以与 DCS 系统融合。

此外,针对武石化现有系统中的一些技术,本文提出了新的优化技术手段。例如使用 SVG 与 AJAX 技术改进 VML 的界面显示;用数据持久层思想改善来自与数据库存取中可能出现的瓶颈,使用 HIBERNATE 框架实现了持久层。这些开放的标准的技术为构建一个开放的 DCS 系统打下了技术基础。

同时本文介绍了如何构建一个压力测试平台，用于测试开发出来的 DCS 演示系统。

关键词：

集散控制系统 面向对象的分析 危险与可操作性分析

## English abstract:

Beneficial from the fast development in science, technology and new energy resources, petroleum industry has made a great progress in China. It is the fact that the petroleum production process is characterized in high temperature, high pressure, flammability, explosive, corrosion and toxic. As the result of this, the petroleum industry has become a high risk industry in accident.

It has been a decade since June, 1998 Wuhan Petroleum Chemical Company started to use the polypropylene product line, which has a production capacity of 10,000 tons per year. In order to reduce the risk during the production process, we are working on a project, which is to integrate the "HAZOP analysis method" into the production process. The project is intended to integrate the "HAZOP analyze method" into the production process. The project suppose to innovate in their current DCS system, use new software techniques to optimize the display part and data processing part in the old DSC and also bring the sensibility to potential hazard during the product process.

In this article, the manufacturing techniques of polypropylene, the present situation of the product techniques and the direction of its evolution are analyzed. I explained what the definition of "HAZOP analyze method" is and how this method can help to reduce the hazard. From the software analysis' aspect, according to the OOA(Object orient analysis), each

constituent part of a typical DCS is analyzed and its brief definition is indicated. According to their functions, a DCS can be divided into several parts as to "Listener", "Measuring Point", "Measuring point list", "Control point" etc. After dividing a whole DSC into individual modules, the "HAZOP analysis method" module could be integrated with the DCS.

Also, taking an aim at some old techniques used in the former DCS, some new suggestions to optimize the system are proposed

For example;

1. The use of SVG and AJAX instead of old VML in user display interface module;
2. The use of "data persistent layer theory" to keep a potential bottleneck in database saving and reading functions, implement
3. The "data persistent layer", implemented by using "HIBERNATE framework".

Those standardized and opened techniques make a comprehensive technical foundation for constructing an opened DSC system.

In the article, As well, how to build a "pressure testing platform" for testing the new system is introduced. This platform also supports the present how the system worked.

**Key words:**

**DCS OOA HAZOP**

## 第 1 章 引言

随着科学技术的进步和资源的不断开发，我国的石油化工得到飞速发展。石油化工由于生产过程具有高温、高压、易燃、易爆、腐蚀、剧毒等特点而成为事故多发行业。在许多重大化工生产事故中，压缩机等设备造成的燃烧爆炸事故占 44.8%，塔、釜、槽等设备造成的燃烧爆炸事故占 27.7%，换热器、废热锅炉等设备造成的燃烧爆炸事故占 20%，管道等设备造成的爆炸、泄漏事故占 8% 以上。这些严重的灾难性事故触目惊心，不但造成巨大的经济损失，而且造成很大的人员伤亡、环境污染，在社会上引起了强烈的反响。为了确保化工、石油化工生产的安全，确保人民生命财产的安全，探讨石油化工生产中可能会发生的事故、采取预防措施尽可能杜绝事故的发生使事故造成的损失减少到最小程度，是一个至关重要且迫切需要解决的课题。

石油化工生产的工艺过程相当复杂，工艺条件要求十分严格，生产装置趋向大型化，以及生产过程的连续化、自动化程度的提高等，介质具有易燃、易爆、有毒、腐蚀等特性。装置中每一条管线，每一台设备，每一块仪表都有其各自的作用。它们在实际操作过程中应始终处在所要求的温度、压力等工艺状态。因设备故障和误操作使生产发生事故如火灾、物理爆炸、化学爆炸、中毒、窒息等的可能性很大，而且造成的危害和损失也极为惨重。这些灾害所造成的严重后果和社会问题远远超过了事故本身。在高科技越来越密集、经济规模越来越宏大的当今，避免

化学工业灾难性事故成为一个国家经济顺利发展的前提条件，也是工业装置平稳安全运行的核心问题。人类文明和社会进步要求生产过程具有更高的安全性、可靠性和稳定性。

因此，本课题针对聚丙烯生产的关键设备如压缩机、换热器等在运行中可能出现的各种异常状态或故障状态通过在线监测，进行及时地、正确地诊断，对设备的运行状态进行必要的指导，以提高生产设备的安全性、可靠性和有效性，减少事故的发生、蔓延与扩大，降低事故的发生率。积极推进应急预案系统的科学化和智能化水平。

## 第 2 章 文献综述

### 2.1 聚丙烯工业、工艺的发展状况

我国聚丙烯的工业生产始于 20 世纪 70 年代，经过 30 多年的发展，目前已经基本上形成了溶剂法、液相本体-气相法、间歇式液相本体法、气相法等多种生产工艺并举，大中小型生产规模共存的生产格局。现在我国的大型聚丙烯生产装置以引进技术为主，中型和小型聚丙烯生产装置以国产化技术为主。国内聚丙烯产量来自乙烯联合生产企业的约占 40%，以炼油厂副产的丙烯为原料的约占 60%。引进技术主要有釜式反应器液相本体-气相本体 Hypol 工艺(三井化学技术)及环管式液相本体-气相本体组合法 Spheripol 工艺(原海蒙特公司，现巴塞尔公司技术)，并在吸收和消化国外釜式液相本体-气相本体工艺和环管式液相本体-气



相本体工艺的基础上，自行设计建成若干套 4 万吨/年 采用 Hypol 工艺和几套 7 万吨/年，采用 Spheripol 工艺聚丙烯生产装置。在 2002 年初，中石化上海石油化工股份有限公司(上海石化公司)采用中国石化自行开发的第二代环管工艺技术，建成 20 万吨/年聚丙烯生产装置。2004 年我国聚丙烯的生产能力已达到 450 万吨每年。陕西省延炼实业集团公司建成一套 10 万吨/年聚丙烯生产装置，中石化、上海石化与 BP 公司合资组建的上海赛科石化有限责任公司新一套 25 万吨/年聚丙烯生产装置，中石油大庆石油炼化公司引进巴塞尔公司技术新建的一套 30 万吨/年聚丙烯生产装置于 2005 年上半年建成投产。截止到 2005 年底，我国聚丙烯的生产厂家已有 70 多家，生产装置共有近 90 套，总生产能力合计约为 545.0 万吨/年，约占世界总生产能力的 11.5%。

鉴于历史上多头引进及引进年代不同，我国聚丙烯工艺呈现多样性，水平参差不齐，总体水平与发达国家相比差距很大等几个特点。我国有使用连续法聚丙烯生产装置 37 套，其生产能力合计约为 447.2 万吨/年，约占国内总生产能力的 82.0%；间歇式生产装置有 54 套，生产能力合计为 97.万吨/年，约占国内总生产能力的 18.0%。

我国聚丙烯生产能力集中在中石化和中石油两大公司。中国石化集团公司生产装置有 46 套装置，生产能力合计为 366.7 万吨，约占我国总生产能力的 67.3%；中石油集团公司生产装置有 31 套，生产能力合计为 151.2 万吨/年，约占我国总生产能力的 27.7%。

## 2.2 聚丙烯生产特点

武石化 10 万吨/年聚丙烯装置于 1998 年 6 月 28 日建成投产，由北京工程公司总承包设计和建设。装置采用国产化聚丙烯环管生产工艺，最初设计年产 7 万吨聚丙烯本色料粒，2004 年改造成 10 万吨的年生产能力，26 个牌号均聚物，设计年操作时间 7200 小时。

整个生产装置由主体生产装置、制氢站和安全消防设施构成。主体生产装置包括 8 个工段：

100 工段：催化剂、助催化剂和添加剂的配制、储存和计量。

200 工段：预接触、预聚合和液相本体聚合。

300 工段：丙烯进料、聚合物脱气及丙烯回收。

500 工段：聚合物汽蒸和干燥。

600 工段：排放系统、废油处理和工艺辅助设施。

700 工段：丙烯保安精制和氢气压缩。

800 工段：添加剂配制、计量和聚合物挤压造粒。

900 工段：产品均化、储存、包装和码垛。

生产工艺具有如下主要显著特点：

1. 采用新型高效载体型催化剂包括球形催化剂

催化剂活性高、定向性好、氢掉敏感、产品等规度易于调节、灰份含量低、熔融指数范围宽。采用球形催化剂可生产球形大颗粒产品。

2. 采用环管反应器进行丙烯均聚反应，物料靠轴流泵进行循环

环管反应器具如下优点：

- a) 结构简单，适于连续生产，运转周期长。
- b) 反应热易于撤除，有效的防止了产生反应器内热点和爆聚等问题。
- c) 反应器体积小，产品牌号转化容易，MFR 相近的产品，转化不需要间隔时间，MFR 相差较大时，转换也仅需数小时。

### 3. 安全系数高，三废排出量少

装置运转期间，可燃性原料储藏量少，并且设计了多重安全消防系统。生产工艺先进，未反应的丙烯循环使用，产品不用脱灰，因而“三废”排放量少。

### 4. 原料消耗低

丙烯单耗不大于  $1.0146\text{tC}_3\text{H}_6/\text{tPP}$

### 5. 设备数量少，工艺流程短

设备材质多为碳钢，建设费用低，建成周期短，设备布置紧凑，占地面积小。

### 6. 采用 DCS 进行操作控制

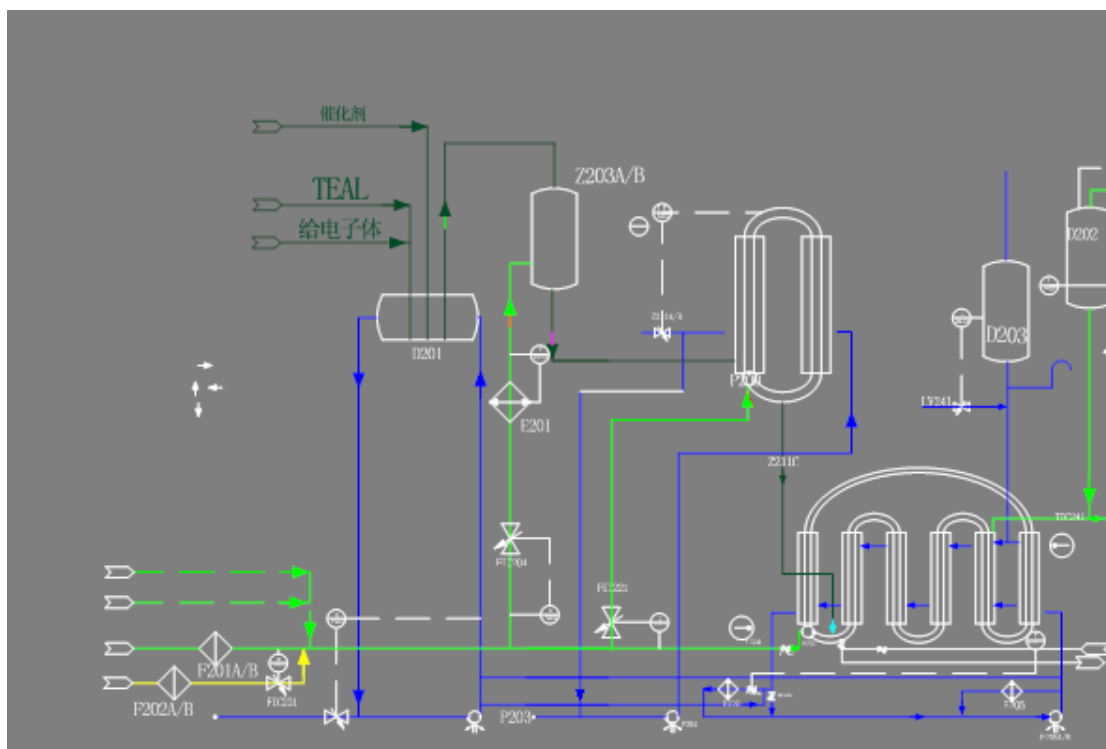
DCS 控制自动化程度高，从而操作人员少，劳动强度低。

### 7. 采用先进过程控制系统

先进过程控制系统优化了操作，使操作更加稳定、简便。

## 2.3 聚丙烯装置概况

武石化聚丙烯反应设备流程图图下：



聚丙烯装置以气体分馏装置分离所得炼厂气中的丙烯为原料，采用国内开发、技术成熟的间歇式液相本体法聚丙烯生产工艺，生产聚丙烯均聚树脂。装置原设计生产能力为  $1.0 \times 10^4 \text{ t/a}$ ，1997 年改造后生产能力达到  $1.2 \times 10^4 \text{ t/a}$ 。该装置工艺过程主要包括原料精制、聚合反应、闪蒸去活和活化再生四个部分，主要设备包括聚合釜、丙烯储罐、活化剂储罐、闪蒸釜，以及丙烯压缩机等。各系统工艺及其操作物料的危险性简介如下。

### 1. 活化剂输送系统

活化剂输送系统操作主要是将活化剂由活化剂运输罐压送至活化剂储罐。系统主要危险物料：

活化剂——三乙基铝 ( $\text{C}_6\text{H}_{15}\text{Al}$ )，为无色透明液体，有强烈的霉烂气味，易燃，化学反应活性很高，接触空气会冒烟自燃。对微量的氧

及水分反应极其灵敏，易引起燃烧爆炸。健康危害：三乙基铝对呼吸道和眼结膜具强烈刺激和腐蚀作用，皮肤接触可致灼伤。

氮气 ( $N_2$ )，不燃，但若遇高热、容器内压增大的情况，有开裂和爆炸的危险。健康危害：氮气为窒息性的惰性气体，空气中氮气过量，使氧分压下降，会引起缺氧。

## 2. 精制系统

精制系统主要将原料丙烯经过脱水、脱硫、脱氧处理，得到合格的精丙烯，供给聚合反应使用。系统主要危险物料为丙烯。

丙烯 ( $C_3H_6$ )，为无色有气味的液体，易燃，与空气形成爆炸性混合物，遇热源、明火有燃烧爆炸的危险。气体比空气重，能在较低处扩散到相当远的地方，遇明火会引起回燃。爆炸极限为  $1.0\% \sim 15.0\%$ 。健康危害：丙烯具有麻醉作用。

## 3. 聚合系统

丙烯在此系统内发生聚合反应生成聚丙烯粉料。系统主要危险物料：丙烯、氢气、聚丙烯。

氢气 ( $H_2$ )，为易燃气体，与空气混合能形成爆炸性混合物，遇热或明火即会发生爆炸。气体比空气轻，在室内使用和储存时，漏气上升滞留屋顶不易排出，遇火星会引起爆炸。爆炸极限为  $4.1\% \sim 74.1\%$ 。健康危害：氢气为惰性气体，仅在高浓度时，由于空气中氧分压低才引起窒息。在很高的分压下，氢气可呈现出麻醉作用。

聚丙烯，为可燃固体，受热分解放出易燃气体。粉料与空气按一定

比例混合达到一定浓度时，如有火花点燃则会迅速燃烧，以至引起强烈爆炸。聚丙烯粉尘云自燃温度为 420℃，爆炸下限为 20g/m<sup>3</sup>。健康危害：本身无毒。

#### 4. 闪蒸系统

闪蒸是接收聚合釜内生成的粉料，并在闪蒸釜内完成闪蒸、置换、去活的操作过程。充内主要危险物料：聚丙烯粉料、丙烯、氮气。

#### 5. 尾气回收系统

尾气回收系统是将气柜内储存的低压丙烯气体加压冷凝为液体。系统内主要危险物料：丙烯。

## 2.4 现代控制系统与在线监测技术的发展状况

技术发展：经过 30 年的发展，现代的 DCS 产品与最初的产品相比，在速度、可靠性、功能、通讯手段等方面取得了巨大的进步。

功能领域的扩展：从功能上现在的 DCS 已经不再局限于控制层，而是增加了更多的管理层功能，向上提供企业管理系统接口。与此同时，DCS 向下不在局限于主工艺装置系统的控制，而是将更多的辅助系统纳入到整个控制系统。

硬件技术的发展：随着硬件技术的发展，DCS 系统的硬件可靠性进一步增加，在系统规模日益扩大的情况下，硬件的平均故障间隔时间（MTBF）也保持在 100,000 小时以上。现在的 DCS 控制站普遍采用 32

位处理器及多处理器技术，在最新的产品上已经开始使用 64 位处理器，相比初期 DCS 控制站的 8 位处理器，运算速度得到大幅度的提升。

最新 DCS 可以划为第四代。第四代 DCS 的最主要标志是两个“I”开头的单词：Information（信息）和 Integration（集成）。

现代的 DCS 系统，逐渐向高集成、一体化、模块化，具有现代理念的过程控制系统的方向发展，它的发展一方面受计算机技术、控制技术、通信技术及显示技术的影响，另一方面又受到生产过程控制和管理要求的驱动。用户希望能将越来越多的功能融合进 DCS 系统中，例如历史数据的分析，决策的支持，安全生产的管理等。

## 2.5 计算机软件发展的趋势

计算机软件技术在近十年中飞速发展，传统软件系统中存在的封闭性，高耦合度，不可移植性等缺点成为了制约软件产品发展的障碍。现代软件技术强调模块化的开发，新的分析思想，如面向对象的分析方法能将复杂的事务抽象成具有特征的模块。随着这种思想的发展，现代软件的组成更加模块化，开放化，标准化。

随着硬件技术的发展和软件关注数据量的不断增大，传统的技术手段在新的软件产品中逐渐出现了瓶颈，非标准化的技术手段逐渐的被标准化的手段取代，这些技术上的进步也将逐渐影响到传统领域的软件产品，对下一代应用系统带来新的变化。

## 2.6 本课题的研究内容与创新点

本课题结合武石化聚丙烯生产线实际设备流程的特点,针对 DSC 系统上位机负责的部分,从武石化现有的 DCS 系统的特点入手,研究化工行业中 DCS 软件的特点,结合现代软件技术发展的趋势,分析目标模型,优化软件系统,从软件系统架构方面改善原有系统。

本课题研究的内容有:

1. 研究组成 DCS 系统的各个功能模块,抽象其功能,用计算机软件编程语言实现这些模块。
2. 研究 DCS 系统中用户界面的设计方案,优化武石化现有系统的界面。
3. 研究 DCS 系统中数据处理的方式,优化武石化现有系统中的数据处理模块。
4. 了解 HAZOP 分析法的思想,结合武石化操作流程,应急预案设计基本的 HAZOP 分析模块,并将其结合进 DCS 系统。

本研究的创新点有:

1. 将面向对象的分析方法分析生产系统,对聚丙烯生产线 DCS 系统进行模块化的描述,以便 HAZOP 分析模块融合进 DCS 系统。
2. 将开放性,标准性的新软件技术引入传统的 DCS 系统中。消除潜在的瓶颈,增强新 DCS 系统的开放性,标准性。



## 第 3 章 DCS 系统的组成和特点

### 3.1 DCS 系统的组成和特点

从 1975 年开始，世界上许多公司都开始研发自己的集散控制系统，总的来说，DCS 系统大致都由现场控制单元、操作管理单元、分散型工业控制网络这三大部分构成。

#### 1. 现场控制单元

现场控制单元由微处理器、存储器、I/O 设备、A/D D/A 转换设备、通信接口等基本部件组成。这些设备可以是单片机、可编程控制器、多功能控制器等等。它们的主要功能是将输入到端子板的过程输入信号发送到控制器上，并把结果输出信号以数字的形式存储与控制器文件区。现场控制单元是实现对现场过程的信号输入、输出变换，过程控制系统数据的采集和各种控制管理的集中控制部件。

#### 2. 操作管理单元

操作管理单元由监视器、微处理器、计算机外设等组成。主要借助计算机软件实现对整个流程的管理和用户操作。对所检测范围内可能出现的紧急情况做出响应，实现被监控系统的最佳。数据库能保存系统运行中的数据以便后期检索或诊断。

操作人员通过操作管理单元提供的界面了解和控制生产流程。典型的

界面包含：整体界面，报警界面，工艺流程界面，各种仪表的状态界面等。这样改善了操作人员的工作条件，提高了工作效率，省去了庞大的仪表操作盘，节约开支，同时在出现异常状态时能更快更准确的调整设备的运行，降低可能出现的风险。

### 3. 分散型工业控制网络

分散型工业控制网络是指现场控制单元、操作单元和管理模块通过通讯链路按照一定的网络拓扑结构而形成的计算机网络系统。分散型工业控制网络传递的信息一般与计算机网络传递的信息不同。DCS 工业控制网络传递的信息主要是使物质或能量产生运动，是一种物质流的信息传递；一般的计算机网络传递的信息是一种按照特定协议通信的情报通讯。

DCS 控制网络的通讯设备采用冗余配置方式来保证其可靠性。网络结构一般采用星形结构，环形结构和总线结构。

### 3.2 HAZOP 分析法

HAZOP 分析法是从过程运行的角度对事故风险进行识别、分析的方法，是 hazard and operability study 的缩写，装置运行偏离正常工况则有可能产生事故风险，由此思路出发研究化工装置运行时灾变过程发生、发展的规律。HAZOP 分析一般是由一个过程设计、实际的运行，以及维护专家组或由 HAZOP 专家系统组成，主要依靠专家知识（包

含实践经验和理论知识), 采用定性分析的方法预测风险。这种方式能成功识别出系统中具有潜在风险隐患的子系统; 然而, 由于主要依靠的是对过程的定性认识, 并不能定量描述灾害发生、发展全过程; 其次, 对于复杂的化工系统单凭专家知识不能有效识别事故在系统中的传播过程, 因为有时一个子系统中被忽略的微不足道的小故障却能逐渐演变为整个系统的灾难; 另外, 基于专家知识的 HAZOP 分析方法并不能提供保证系统安全的具体措施。所以建立基于化工过程事故模型的 HAZOP 分析方法有重大的理论意义和实用价值。

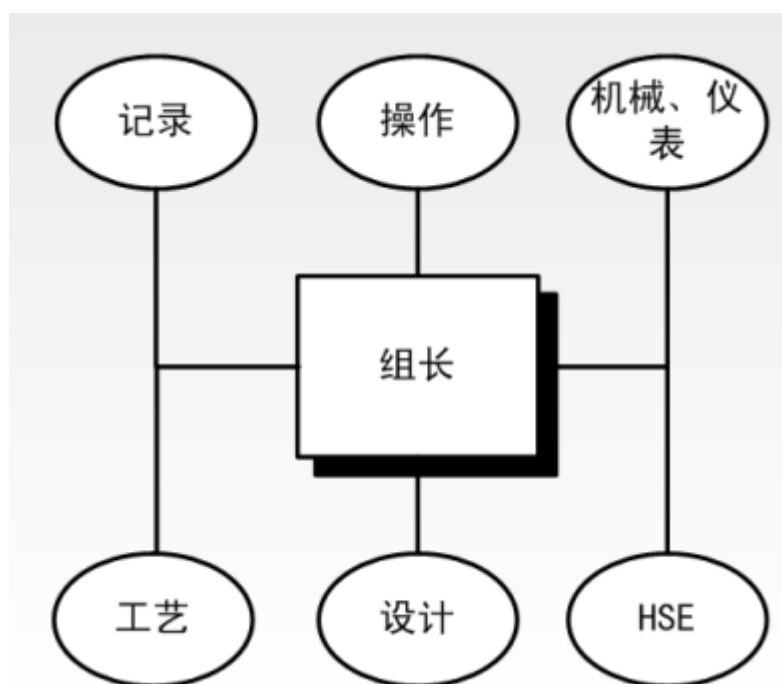
HAZOP 研究是以系统工程为基础的一种可用于定性分析或定量评价的危险性评价方法, 用于探明生产装置和工艺过程中的危险及其原因, 寻求必要对策。通过分析生产运行过程中工艺状态参数的变动, 操作控制中可能出现的偏差, 以及这些变动与偏差对系统的影响及可能导致的后果, 找出出现变动可偏差的原因, 明确装置或系统内及生产过程中存在的主要危险、危害因素, 并针对变动与偏差的后果提出应采取的措施。

在武石化聚丙烯生产线中有针对各个工段的安全预警操作手册, 手册中描述了若出现了异常后, 应当如何操作控制设备消除危险的方法。系统需要将这些手册中的经验和处理方法数字化, 层次化, 逻辑化, 并存储进数据库以便查询。

当数据处理模块发现测量到的数据超出了界限值后, 需要启动 HAZOP 分析法, 结合转换好的预处理手段流程 (来自武石化操作手册), 用最快的速度给出处理意见, 必要的时候直接开始启动控制设备干预生产, 降

低危险发生的可能。

HAZOP 由一位经验丰富的主持人提供指导。对于石油和天然气项目，通常将会组织一个核心小组，包括流程、仪器仪表、机械、工程施工和运营人员，还可以在工作的某些阶段，根据需要加入流程技术人员、环境专家和公司健康、安全和环境（HSE）员工。



武石化 10 万吨聚丙烯生产线有专门的操作手册，描述了生产过程中可能遇见的问题和处理方式，但是这些纸质的文档查询阅读起来比较繁琐，除非是富有经验的操作员能将所有的细节烂熟于心。但是可以想象，当事故发生时，人本能的惊慌会使思考出现问题。结合武石化安全生产应急预案，软件系统可以实现将这预案，手册中的内容数字化，层次化，逻辑化。并用 HAZOP 分析算法事先建立好风险模型，由计算机担当定期

检查反应状态的工作，快速的有预见性的发现风险。

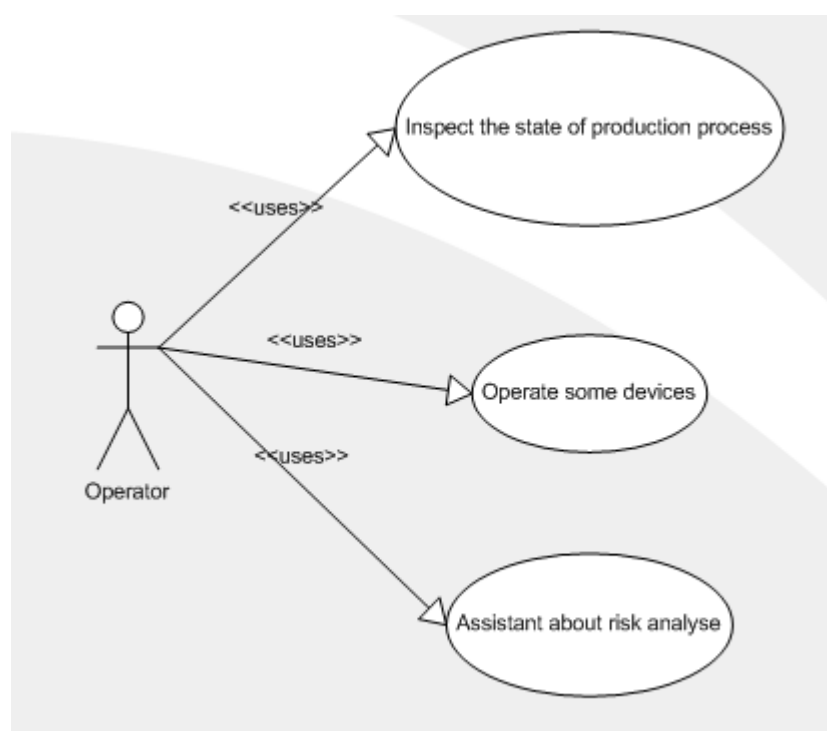
### **3.3 本章小结**

本章介绍了 DCS 系统的特点和发展，DCS 系统实际上是工业控制与计算机软件的结合。将传统生产中的操作流程，安全应急预案，结合 HAZOP 分析法，事先建立风险评估模型，融入 DCS 系统，能在运行的角度中发现潜在的问题，降低发生事故的风险。

## **第 4 章 系统分析与设计**

### **4.1 系统需求分析**

DCS 系统是生产线技术管理人员了解生产情况，调整设备的一个工具，对于用户（生产技术管理人员）来说，他们需要能了解、控制生产情况的系统，此外他们也需要有智能预警，降低风险的辅助功能。Use Case 图如下：



## 4.2 DSC 系统的架构和通信方式

在 DSC 系统中，按照功能和所处位置、操作人员不同划分，可以分为：控制站，操作站和工程师站，

DCS 操作站实际上就是典型的计算机，有着丰富的外围设备和良好的人机界面，70 年代中期，主要是在原小型计算机基础上开发的，此时 CRT 阴极射线管技术已成熟，但外部存储器温氏硬盘技术至 80 年代才普及，所以第一代操作站采用 CRT 技术为特征。第二代操作站以采用温氏硬盘为特征，同时第二代 DCS 操作站还具有如下特征：操作站和工程师站（或称工作站）分开，也有产品将操作站的历史数据存储用硬盘（历史模件）和高级语言应用站（应用模件）分别独立挂在通信网络上。

操作系统方面除了采用 DCS 专用系统以外,有的产品采用 Unix 等操作系统;实时数据库的性能逐渐完善;在人机界面即流程图画面方面,也由半图形方式、全图形方式过程为目标图形方式,以 GVI 图形用户界面为平台,并采用鼠标,而且在工程师站中进行控制策略组态和制作流程图画面采用菜单、窗口、CAD 技术等,使人机界面友好。第三代 DCS 操作站是在个人计算机(PC)及 Windows 操作系统普及和通用监控图形软件已商品化的基础上诞生的,面对用户要求的 DCS 系统应具有开放性,便于系统集成各操作,与 Windows 技术其一致等。目前大多数 DCS 操作站和工程师站已采用高档 PC 机或工控机,Windows NT 等操作系统、客户机/服务器(C/S)结构、DDC(动态数据交换)或 OPC(用于过程控制对象链接嵌入)接口技术,通过以太网接口与管理网络相连,逐渐淘汰了传统的通信总线。用户显示方面,系统通过监控图形软件开发环境,有的厂家以此为平台,形成“软 DCS”操作站,但这些系统多用于中小型 DCS 系统,又或以此类软件为核心,进行二次开发;有的厂家对原来的组态软件进行改造,使之符合上述特点,满足系统开放要求。系统的开放成为了未来 DCS 系统发展的一种趋势。

操作站要实现其多项功能,必须完成数据组织和存储两方面任务,如与工位号相关的一些数据,在操作站中要对由某控制站某端子与现场仪表相连的,由生产线中的物理位置决定的工位规定工位号(即特征号或标签 Tag)各工位说明(可以用汉字或其他符号),使之与工艺对象一致,以保证工艺操作人员的操作,工位号可以在整个系统中通用。其它还有

系统配置、操作标记、趋势记录、历史数据管总貌画面组态、工艺单元或区域组态等，这些均组织成文件，最终形成数据库，存储在硬盘的相应区域，使数据具有独立性和共享性、保证数据的实时性、完整性和安全性。

控制站是整个 DCS 的基础，它的可靠性和安全性最为主要，死机和控制失灵的现象是绝对不允许的，而且冗余配置、设备的掉电保护、抗电磁干扰、构成防爆系统等方面都有效而可靠，才能满足用户要求。以后 DCS 厂家在通信网络、操作站、工程师站几方面制作逐步开放或可以互换并以提供"解决方案"为主时，作为制造厂，仍应保持控制站的软硬件的生产专有，因为这是传统 DCS 或称常规 DCS 的根本。在完善的 DDS 技术中，对 A/D、D/A 转换及控制算法，分别引入扫描周期和控制周期概念，在第二代、第三代 DCS 控制站中，扫描周期可以比缺省值 1 秒更短（1 秒对一般温度的等物理量的测量控制是足够的），如可以选用 0.2-0.5 秒，以满足少数反应快的控制对象的要求。

现代 DCS 系统逐渐采用了工业以太网作为通信载体，武石化聚丙烯采用的是传统的 485 总线实现 DCS，本设计关注的是操作站和工程师站所完成的工作，在设计通信方式时采用标准的 Socket 通信方式进行，使用 TCP 协议作为通信协议，将各个模块之间需要传输的数据按照约定的格式进行封装后传输。

#### 4.3 系统整体设计



#### 4.3.1 为何要使用面向对象的分析方法分析 DCS 系统

武石化现有 DCS 系统采用的传统的 DCS 软件，整个系统较为封闭，在这样的系统中想要加入新的模块存在一定的难度。

OOA（面向对象的分析）模型由 5 个层次（主题层、对象类层、结构层、属性层和服务层）和五种活动（标识对象类、标识结构、定义主题、定义属性和定义服务）组成。在这种方法中定义了两种对象类之间的结构，一种被称为分类结构，另一种称为组装结构。分类结构就是所谓的一般与特殊的关系。组装结构则反映了对象之间的整体与部分的关系。

使用面向对象的分析方法有如下好处：

1. 加强了对问题域和系统责任的理解；
2. 改进与分析有关的各类人员之间的交流；
3. 对需求的变化具有较强的适应性；
- 4 支持软件复用。
5. 贯穿软件生命周期全过程的一致性。
6. 提高软件的实用性；
7. 利于用户参与对系统的定义。

OOA 分析法能将复杂的系统模块化，这样各个模块的聚合就可以形成复杂的 DCS 系统，同时新模块也能较好的与其他模块融合。

#### 4.3.2 DCS 系统功能模块分析

研究中与武石化负责安全维护的主工程师进行了多次沟通，详细询问了负责安全的部门是如何应用软件系统进行安全管理和预警的，重点对 DCS 系统的组成进行了信息交流。之后将收集到的要点进行抽象和归纳后，再次与其进行了反馈。最终得出的分析结果结合武石化公司制定的人为的管理手段和实际工作流程，对其 DCS 系统形成了模块的定义和划分。

根据对系统主要功能，特点和实际生产流程中管理方式，对目标系统的功能模块进行了如下的定义：

1. 总体的结构模式是由若干个监测设备实时监测目标设备的运行状态，监测的数据经过归组、记录，若出现测量值超出工艺流程中规定的阈值，则发送控制信号给控制设备进行干预。
2. 用于负责监听来自于各个测量点的数据的模块叫做监听器，这个模块的程序运行于整个生产过程，负责监听来自于指定监听模块的数据。
3. 用于处理数据的模块叫数据处理模块，监听器收到的数据转交给独立的数据处理模块，对数据进行分组和存储，若在处理数据时发现异常，数据处理模块将进入 HAZOP 分析，并将得到的结果分解成指令信号组，通知控制点设备执行指令。
4. 用于测量设备中的压力、温度、流量等数据的数据处理模块叫监控点对象。监控点对象接受来自于工业传感器，PLC 或单片机设备采集上来的数据，并将数据整理之后按照预先配置的规则发送给监听

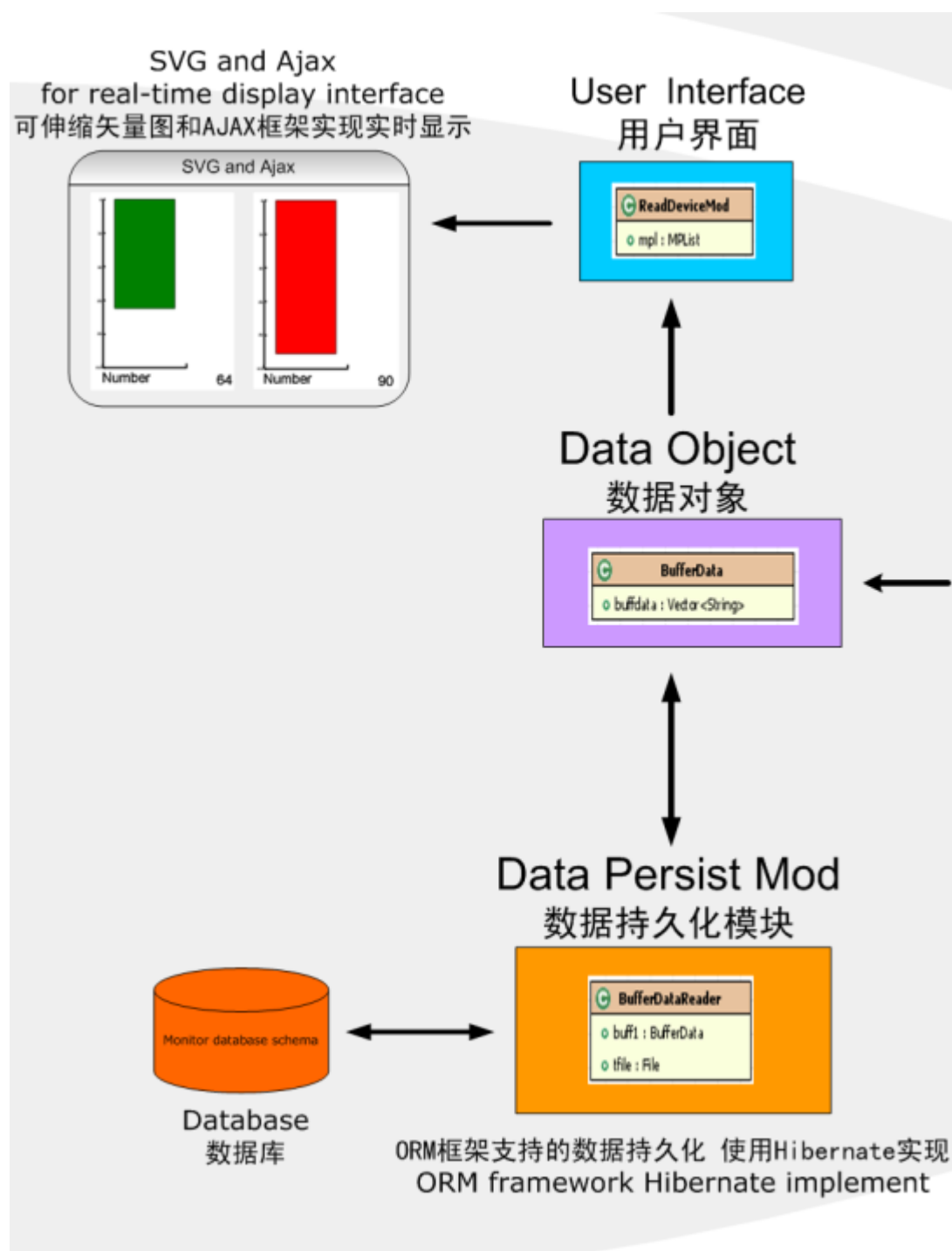
器。

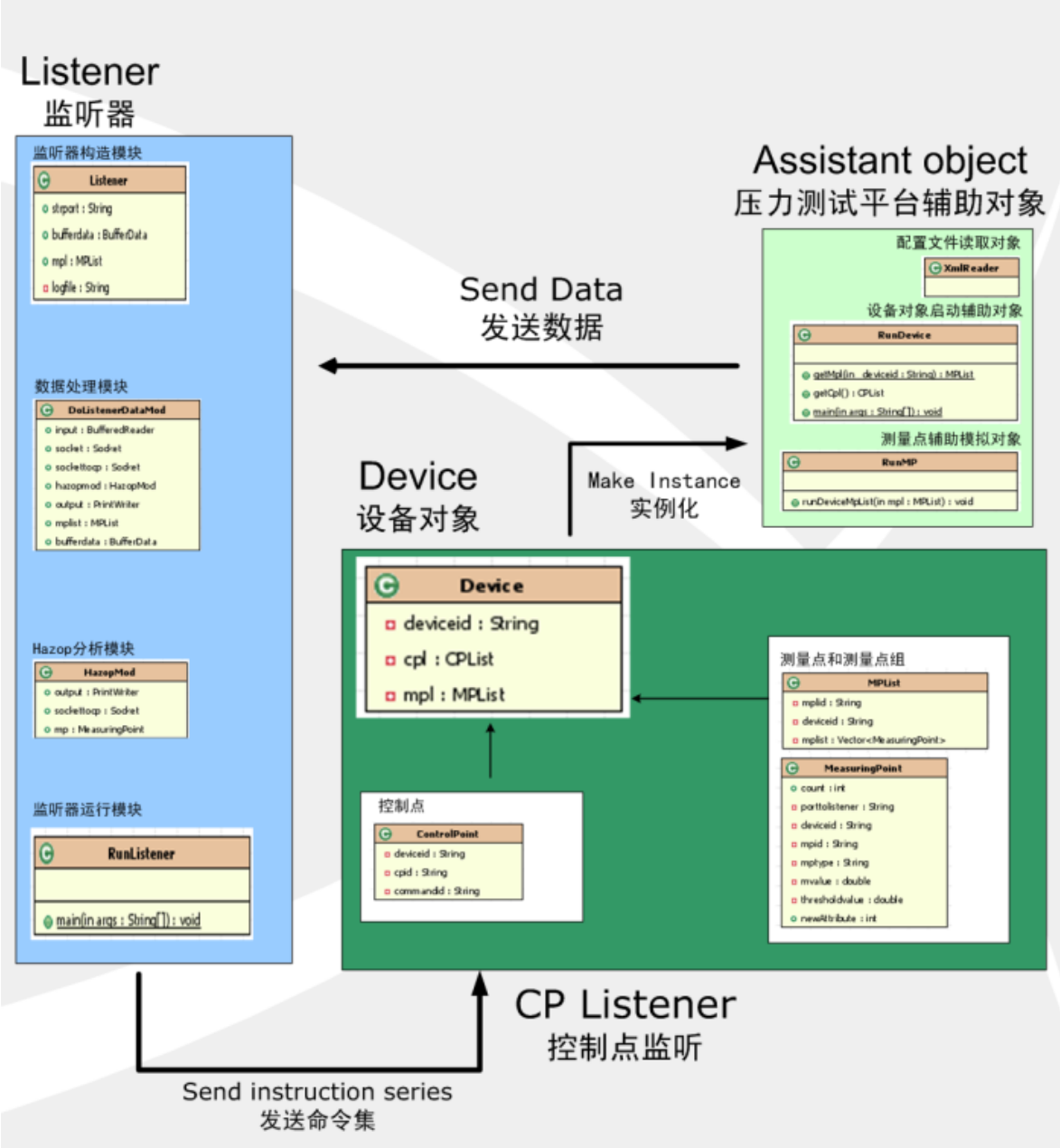
5. 用于接收来自于数据处理模块发送过来的指令信号组的模块叫做指令处理模块。指令处理模块用于接收来自于数据处理模块发送过来的指令数据报文,并将其转换成控制设备执行机构所能理解的数据格式,通知执行机构进行控制。同时指令处理模块需要保持和中央处理模块的联系,保证指令正常无误的执行。
6. 用于进行 HAZOP 分析的模块叫做 HAZOP 处理模块。HAZOP 处理模块用于在线处理和分析 DCS 系统收集到的各监控点采集到的数据,将处理意见提交给数据处理模块。
7. 将反应系统的历史运行信息储存起来的模块成为数据持久化模块,数据可以被存储进数据库,也可以存储进文件。被存储起来的数据为今后的 HAZOP 分析或技术改进,设备检修提供数据支持。
8. 用户读取信息,输入操作指令的模块成为用户界面模块。用户界面模块需要形象的展示出系统中各个模块的运行状态,并且在当异常发生时用声音,闪烁等方式提醒操作人员注意。

通过对整个 DCS 系统功能模块的分析,部分模块间的数据可以通过共同的内存空间实现,不处于同一内存空间的模块间可以用 SOCKET 方式进行通信。

监听器,数据处理模块,辅助模块成为中央处理单元。测量点,控制点成为现场处理单元。

以下两张图为系统的整体模块分布图:





接下来结合现代软件面向对象思想，对各个模块进行分析和设计。

### 4.3.3 开发环境的确定

从通信特点来考虑，DCS 系统上位机部分关心的重点的数据通信，数据处理和数据展示，所以在软件系统上应该采用便于网络通信，数据处理的语言和技术。从系统分析方面考虑，在分析 DCS 系统时运用了面向对象的模块化分析，所以使用一个面向对象的开发语言能很好的维持对象模型。

DCS 系统要求运行在稳定、安全的操作系统中，UNIX，LINUX 操作系统成为了最佳选择，进程管理机制，文件系统，内存管理机制能对工业级的软件提供良好的保证。

结合这些特点，系统选定使用 JAVA 为编程语言，LINUX 操作系统为开发用环境。这样的好处是：能保证软件的健壮性，操作系统无关，并且有大量的开放的框架和资源可以应用。

#### 4.4 监听器的设计

监听器是整个系统的核心，它在系统的运行中负责对来自各个测量点数据的收集、分析、存储。监听器按照生产流程的需要可以有多个，但是每一个监听器至少要负责监听具有相似特性的数据。在性能上，监听器必须对每一个连接上来测量点分配独立的内存空间，并且要能并行的处理这些数据。

监听器应该是一个多线程的监听程序，鉴于化工反应的特点，采用有连接状态确认的 TCP 协议保证和监控点的联系。在 DCS 系统工作的过程

中，监听器和测量点之间的数据连接是不会中断的。但是在工作中难免会出现通信异常，监听器中存在状态监测功能，当发现连接中断后会给予管理员提示。

监听器根据配置文件中的描述，指定监听端口。当监听器接受来自测量点的连接后，启动新独立的数据处理模块线程，专门处理来自于该测量点的数据。

以下代码展示了监听器模块的部分属性和方法：

```
public class Listener implements Runnable{
    String strport;
    BufferData bufferdata = new BufferData();
    MPLList mpl = new MPLList();
    . . . . .

    public void setLogfile(String logfile) {
        this.logfile = logfile;
    }

    public void run(){
        run(strport);
    }

    public void setPort(String port1){
        strport = port1;
    }

    public void run(String instrport) {
        . . . . .
    }
    public String getStrport() {
        return strport;
    }
    public void setStrport(String strport) {
        this.strport = strport;
    }
}
```

代码对监听器进行了定义和描述，监听器对象实现了 `Runnable` 接口，是线程类的具体实现。监听器有自身的属性，如 `strport`，描述的是通信端口；有测量点组对象，如 `mpl` 等。同时监听类还包含了很多方法，用于实现不同的功能。

## 4.5 数据处理模块

数据处理模块实现线程接口，通过 `SOCKET` 获得来自于测量点的数据报文，解拆后更新内存中对应的数据对象。`HAZOP` 模块也是由数据处理模块实例化。测量到的数据经过规整更新之后，交给 `HAZOP` 模块判断是否超过预设的阈值。

### 4.5.1 数据处理模块中的内存处理算法

数据处理模块是由监听器实例化的，只有在监听器收到数据包后，对应的监听器线程才会开始运行一个新的数据处理模块线程。数据处理模块线程最初是没有给测量数据分配内存空间的，每一个监听器在初始化的时候会创建一个测量点组对象，这个对象通过 `SET` 的模式共享给数据处理模块。

当监听器接受了一个 `SOCKET` 的连接后，这个 `SOCKET` 将被 `SET` 进数据处理模块，并由数据处理模块开始对 `SOCKET` 中的数据进行分析和



处理。这样做的好处是降低监听器的负载，使得不同测量点的数据在内存空间中相对独立。

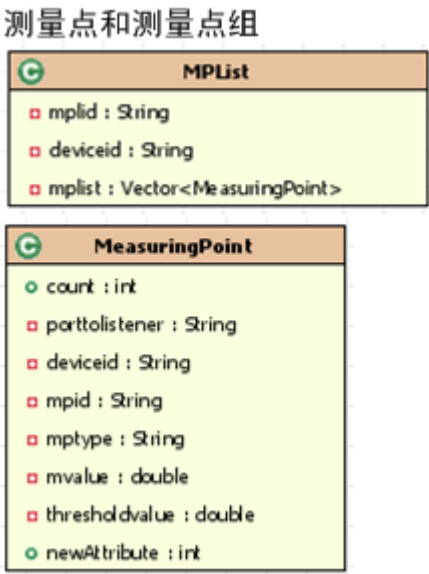
#### 4.6 测量点对象的设计

测量点对象在实际系统中对应设备中的各个测量设备，例如温度传感器，压力传感器等。测量点是被包含于设备中的，每个测量点用设备编号同设备关联起来，而每个测量点自身又有独立的编号。编号的依据参照武石化实际设备编号设置。这样的编码方式能保证测量点和设备之间的逻辑层次关系。

测量点的初始化参数由配置文件决定，与监听器的联系采用 TCP 协议连接，双方采用约定的报文格式通信。

在实际系统中，测量点需要同下位机通信，下位机是指用于测量的各个传感器或单片机，测量点的另一个重要作用是将下位机收集到的数据重新整理归组，并且按照报文的规则发送至监听器。

同属于一个设备的测量点具有相同的设备编号，在实际设计中可以建立新的数据结构—测量点组。测量点组是属于同一设备的部分测量点的集合，但是和其他测量点组在观测特点上又存在一些不同。创建测量点组这样的结构的好处是能表现出测量点集合在工段中的特点。测量点和测量点组的类成员如下图：



#### 4.7 数据结构设计

上位机是指：人可以直接发出操控命令的计算机，一般是 PC，屏幕上显示各种信号变化（液压，水位，温度等）。下位机是直接控制设备获取设备状况的计算机，一般是 PLC/单片机之类的。上位机发出的命令首先给下位机，下位机再根据此命令解释成相应时序信号直接控制相应设备。下位机不时读取设备状态数据（一般模拟量），转化成数字信号反馈给上位机。简言之如此，真实情况千差万别不离其宗。上下位机都需要编程，都有专门的开发系统。

系统在各个部件间通信的过程中，下位机的传感器与测量点模块之间，测量点模块与监听器之间的通信均采用 `socket` 方式通信，所以各个模块之间需要事先定义好数据包格式。

下位机和测量点之间采用如下数据报文格式：

“设备 ID+测量点 ID+测量点类型+测量数据+校验位”

数据报文确定后，将对其做进一步扩展和封装，最终形成一个独立的对象，该对象将在系统中被不同的模块读取。采用自定义的对象封装数据，有利于今后对其持久化，同时遵从了系统设计中的模块化设计要求，若日后数据报文格式需要发生变化或要求在系统中扩展其他的信息，这种对象化的数据设计能快速准确的完成数据的再定义。

## 4.8 设备对象的设计

按照面向对象分析法（OOA），可以将设备对象抽象成若干子对象的组合，

即：设备 = 设备属性参数 + 测量点组 + 控制点组

用 JAVA 代码表示如下：

```
public class Device {  
    private String deviceid;  
    private String devicesn;  
    private CPList cpl;  
    private MPList mpl;  
    private Set mps;  
}
```

以上的代码对设备对象进行了定义，设备是由先前定义好的测量点，控制点和自身的属性聚合而成的。

系统对“设备”的定义由三个部分组成，设备属性参数是一组描述设备特性的数值，例如设备类型，规格型号，设计温度，设计压力等，但是这些数据本身并不参与 DCS 正常的控制流程。测量点组是属于同一

个设备的测量点集合，控制点组是属于一个设备的控制点集合。设备和设备之间用设备工艺位编号进行区分，此工艺号来自于实际生产流程中的编码。

“设备”这个对象在软件系统中实际是一个抽象的集合，并不是实际存在的对象。建立“设备”这个对象的好处是能将各个子对象归结为一个集合，同时使软件系统中的对象更贴近于实际事务，使其两者之间的关系更将紧密联接。号或设备编号区别。

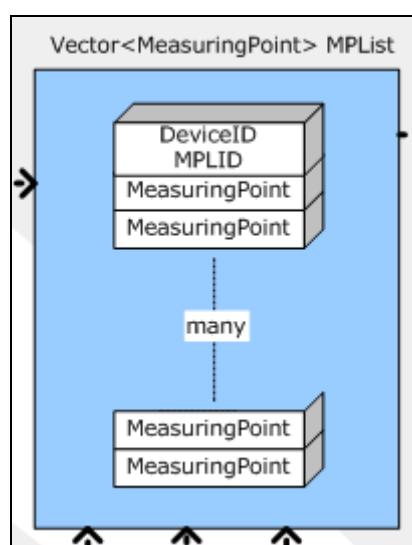
参照武石化设计生产线情况，整个聚丙烯生产流程中，按照设备类型可以分成以下几类。

名称	包含类型
安全阀	控制点
反应类设备	测量点
化工机械类	控制点
换热设备	测量点
机泵、压缩机	测量点
容器	测量点
过滤设备	测量点
分离类设备	测量点

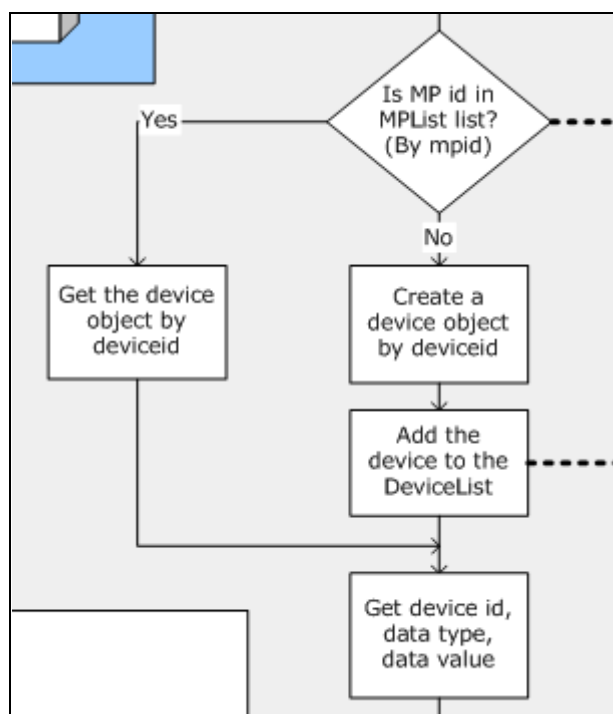
## 4.9 系统的内存管理

在系统运行中，主要有两的大的内存区域，分别是测量点组和自定义的 BUFFERDATA 对象。

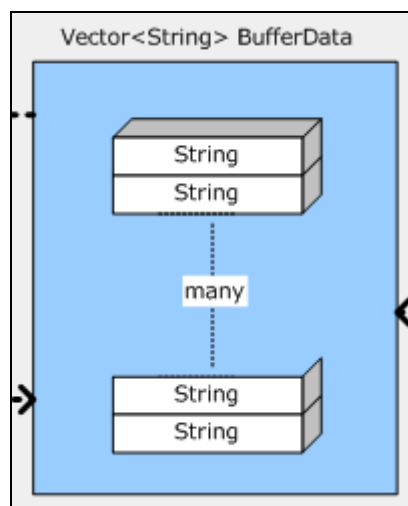
测量点组在前面的章节中已有详细的介绍，在系统运行中，测量点组在内存中是始终存在的对象，由于在系统分析的过程中本着抽象对象模型的思想，在系统运行中，可以认为有若干个虚拟存在的设备传感器对象存在。每个监听器有自己对应的测量点组对象，通过设备 ID 值区分。其数据结构如图：



当一个新的测量数据被接受到后，程序查询现有的测量点组，若没有发现符合的测量点记录，则在测量点组中增加一个新的测量点对象；若发现已有该测量点对象，则更新对应的数据。测量点组中保存的数据始终是这个测量点最新一次的测量数据。其逻辑流程图如下：



自定义的 BUFFERDATA 对象实际上是为了针对系统的日志记录设置的，它实际上是对 JAVA 中的 VECTOR 对象的一次封装，加上了自定义的属性和方法，形成的数据对象。其数据结构如图：



系统在运行时可能会遇见这样的问题，当测量点数量非常多，数据量非常大的情况下，有可能会造成数据处理模块的响应不及时，造成通讯的拥堵或进程的崩溃。为了避免这种情况，定义了 BUFFERDATA 数据对象，

它的主要是起到一个缓冲区的作用，接受到每条测量点数据先存放在内存空间中，单独的线程 `BufferDataReader` 按照设定的时间间隔将 `BUFFERDATA` 对象中的数据记录到日志媒体中，并清空 `BUFFERDATA` 对象。

#### **4.10 本章小结**

在这一章中，主要阐述了如何应用面向对象的分析方法分析 DCS 系统中的各个部件。将整个生产装置中按照由大到小的顺序分为：设备，控制点，测量点组，测量点几个部分。结合软件设计的需求，分析描述了主要对象在软件中需要保证的特点，为下一步对每一个具体模块的设计打下了基础。

### **第 5 章 聚丙烯装置的 HAZOP 模块的设计**

#### **5.1 控制点对象的设计**

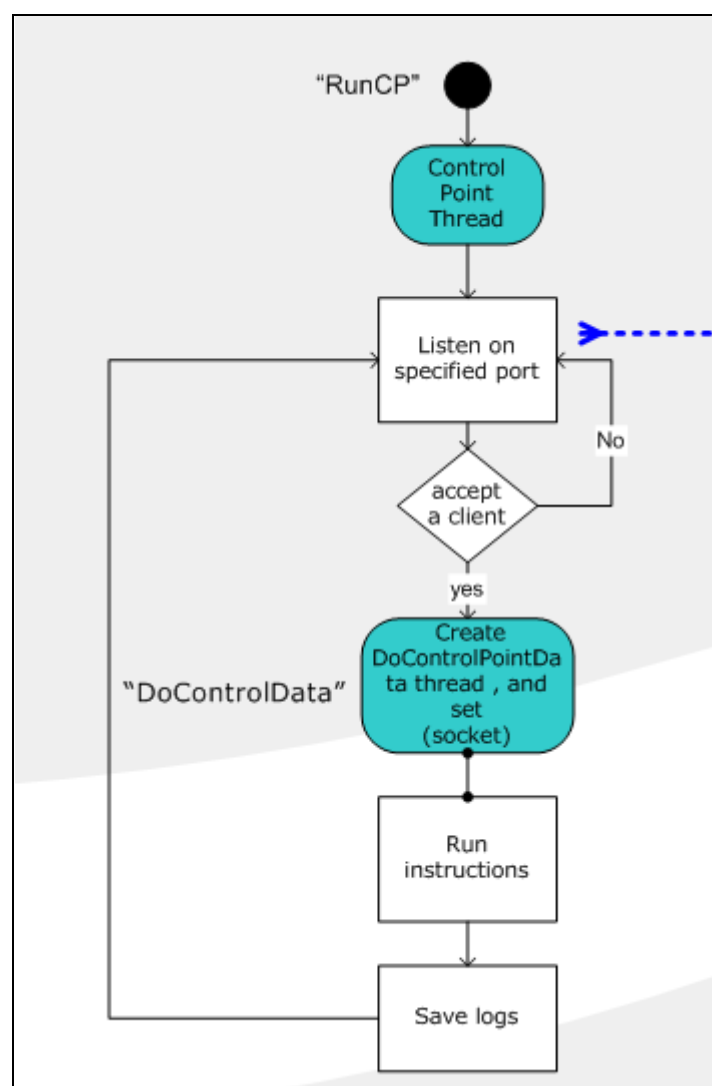
控制点对象在实际系统中对应设备中的各个控制设备，例如阀门，泵等装置。控制点同样被包含于设备中（与测量点相同），通过设备编号和控制点编号维持层次关系。

与测量点不同，控制点需要接受来自中央处理模块的指令，并且需要将执行后的情况通知中央处理单元。通信过程要求较高的响应速度和稳定性。所以在控制点上要运行监听程序，以便随时响应来自于中央模块的指令。

当系统的各个模块负责完成不同的功能，测量点对象在实际系统中对应设备中的各个测量设备，彼此之间独立，但是又相互依赖。作为一个整体的系统，需要保证在运行中不出现死锁，内存泄漏等问题。

所以在系统的设计中，一切事物和数据都采用面向对象的思想进行建模，在具体设计中，参考经典设计模式中的模型进行设计。例如在多线程管理方面，采用 Factory 方式设计。

控制点对象的工作流程图如下所示：





## 5.2 数据的收集和处理

监听器的工作相对单一，但是需要同时管理来自多个测量点的数据，从模块的功能设计角度上讲，应该遵从尽量简化、单一化模块功能的思想。当监听器接受了一个来自测量点的 Socket 通信请求后，将实例化新的模块对接收到的数据进行处理。

### 5.2.1 数据的收集

化工反应中，从测量点收集到的数据是相对离散的，数据类型一般包括：温度，压力，流量，液位高度，流速等。

系统中引入了数据处理模块这个对象。数据处理模块在系统最初运行时是没有实例化的，当监听器接收到数据报文后，将实例化新的数据处理模块，并将其加载进独立的新线程中。这样做的好处是可以保证来自不同测量点的数据被处理时是相互独立的，不会因为某测量点的异常造成整个监听器的工作停止。

数据处理模块主要对接受到的数据报文进行拆解、分析、归档和实体化，每一个数据报文中均会包含来自测量点的 ID 值，数据类型描述，数据值等内容，数据处理模块需要将这些数据从报文中分析得到，并创建或更新对应的内存空间。

### 5.2.2 数据的持久化

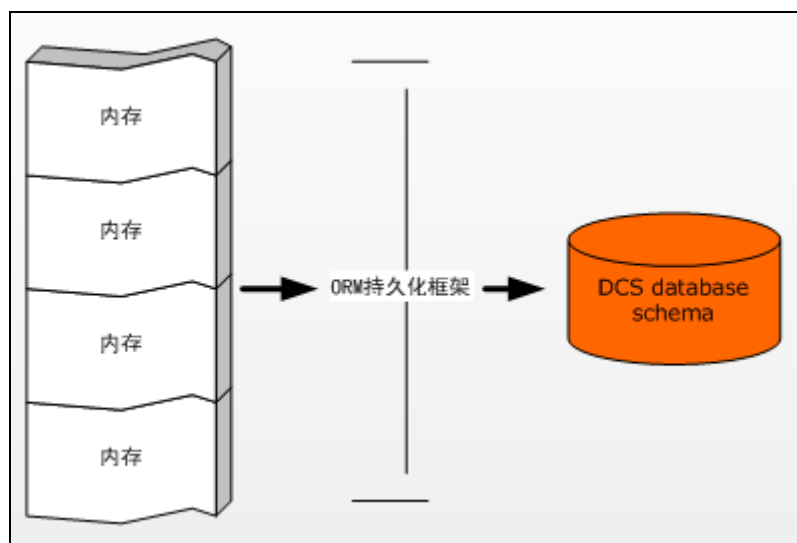
数据持久化顾名思义就是把程序中的数据以某种形式保存到某存贮介质中，以达到持久化的目的。当程序运行时，一些数据是临时保存在内存中，一旦退出系统，这些数据就丢失了。那么，使用某种手段将数据保存在硬盘上或者数据库中，这样即使退出系统后又重新启动系统，那么这些数据仍然可以重新找回来。

例如监听器向某个设备中的某个测量点添加了一次采样得到的数据，那么这个系统需要将新添加的数据保存到数据库中，否则系统退出或者电脑重启后该采样数据就会丢失。将数据从内存保存到数据库中，这便是数据的持久化。当然，数据库只是持久化方式中的一种，也可以保存在其他的永久存贮介质中。

好的持久化框架应具有以下特点：

1. 性能：高性能，最好能直接使用具体数据库特性
2. 通用：可移植，在不同数据库之间移植
3. 方便：使用方便，o/R mapping 非常方便
4. 安全：事务机制好，很好地支持 CAID 等。

图为数据持久化的过程示意图。



在工业控制领域，由于数据的变化比较频繁，而且要求能快速的读取和分析，如果使用传统的软件架构，将数据的存储放置在数据库中，将可能会造成系统的数据访问瓶颈。数据在内存中的访问比通过数据库要快捷的多。但是单纯的放在内存中，一旦出现系统掉电或者程序运行异常，将丢失内存中所有的数据。所以，数据不能单纯的存在于内存中，必须按照一定的方式进行保存。

将数据持久化思想引入 DCS 系统，相对传统方式能提高效率，同时也能保证对数据稳定性。

实现 ORM 框架的有很多项目，如 Entity Enterprise Java Bean, Java Persistence API, Hibernate, TopLink 等。

系统设计中采用 Hibernate 框架实现 ORM。

由于整个系统各个模块是按照面向对象的思想进行设计，所以能水到渠成的应用数据持久化框架来处理数据的保存问题。使用持久化框架可

以避免数据库平台的依赖，降低数据库带来风险，便于系统的后续开发和代码的可重用性。

数据经过数据处理模块更新至内存对象后，接下来将被持久化，数据库中有相应的表对应测量点和设备，最新的采样数据将被持久化对象更新至数据库，以便界面读取显示。

### 5.2.3 Hibernate 持久层的实现

将系统中的各个模块进行 ORM 映射，设备与测量点组之间是一个 1 对多（1..n）的关系。采用 Hibernate 中推荐的带中间表的 1 对多映射方式为两者建立映射关系：

以下为 Device 对象的映射文件：

```
<hibernate-mapping>
  <class name="HAZOP.Device" table="device" catalog="HAZOP">
    <id name="devicesn" type="java.lang.String">
      <column name="DEVICESN" length="50"/>
      <generator class="assigned" />
    </id>
    <property name="deviceid" type="java.lang.String">
      <column name="DEVICEID" length="50" />
    </property>

    <set name="mps" cascade="all"
table="devicemeasuringpoint">
      <key column="devicesn" />
      <many-to-many column="mpsn" unique="true"
class="HAZOP.MeasuringPoint" />
    </set>
  </class>
```

```
</hibernate-mapping>
```

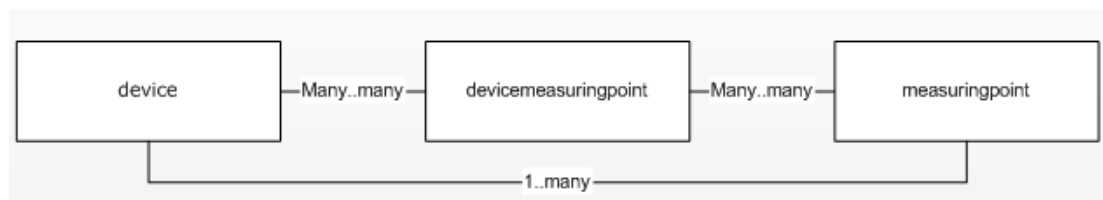
以下为 MeasuringPoint(测量点)对象的映射文件:

```
<hibernate-mapping>
  <class name="HAZOP.MeasuringPoint" table="measuringpoint"
catalog="HAZOP">
    <id name="mpsn" type="java.lang.String">
        <column name="MPSN" length="50" />
        <generator class="assigned" ></generator>
    </id>
    <property name="mpid" type="java.lang.String">
        <column name="MPID" length="50" not-null="true" />
    </property>
    <property name="deviceid" type="java.lang.String">
        <column name="DEVICEID" length="50" />
    </property>
    <property name="mptype" type="java.lang.String">
        <column name="MPTYPE" length="50" />
    </property>
    <property name="mvalue" type="java.lang.Double">
        <column name="MVALUE" precision="15" scale="3" />
    </property>
    <property name="thresholdvalue"
type="java.lang.Double">
        <column name="THRESHOLDVALUE" precision="15"
scale="3" />
    </property>
    <join table="measuringpoint" optional="true"
inverse="true">
        <key column="MPSN" />
        <many-to-one name="device" column="devicesn"
not-null="true" />
    </join>
  </class>
</hibernate-mapping>
```

映射文件中将数据库表中的字段和 JAVA POJO 对象中的对应字段进行了映射, 例如 Measurepoint 中的 mpid 对应数据库中的 MPID 列, 类型为 String。此外在映射文件中还表现出了表和表之间的约束关系,

系统中使用的是带中间表的一对多映射。

在数据库中，建立三张表：device 表，measuringpoint 表和中间表 devicemeasuringpoint，他们的关系如下图所示：



经过这样的设置，实际上将程序中的对象与关系数据库中数据源进行了映射，跨越了面向对象与关系数据库之间的差异。

#### 5.2.4 连接池的使用

Hibernate 与数据库的连接若不加以控制，在系统运行长时间后可能会出现无法创建新连接的情况，在多线程协同工作的系统中，这个问题尤为突出。在这里，为持久层加入了连接池管理，确保与数据库的连接稳定安全。

系统是使用的是开源产品 proxool 连接池。

Proxool 是一个基于 JAVA 的连接池产品，对于 proxool 在文章中不再做过多的介绍，详细的文档可以在它的发布网站 <http://proxool.sourceforge.net/> 中可以找到。

以下是连接池配置文件 proxool.xml 中的片段：

```
<house-keeping-sleep-time>90000</house-keeping-sleep-time>
<maximum-new-connections>30</maximum-new-connections>
```

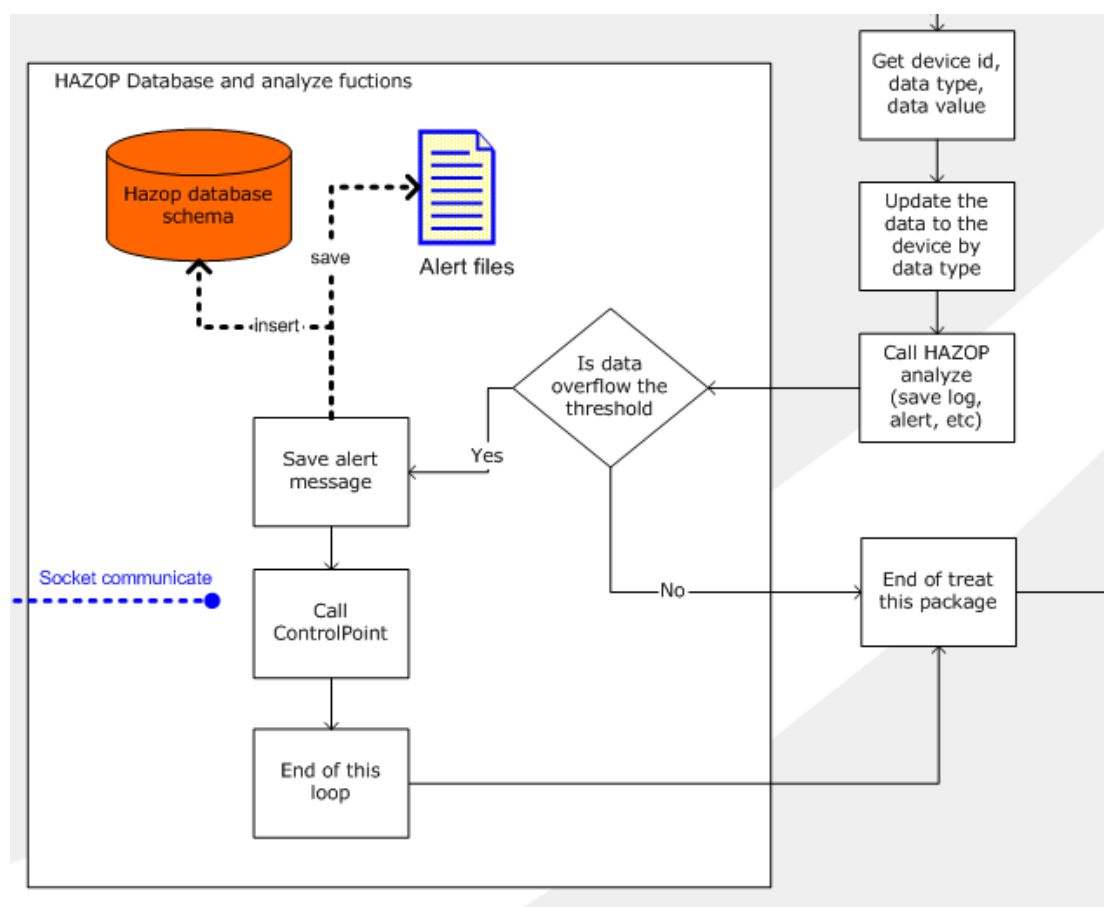
```
<prototype-count>5</prototype-count>  
<maximum-connection-count>100</maximum-connection-count>  
<minimum-connection-count>10</minimum-connection-count>
```

通过对连接池参数的调整，保证整个系统最多产生 30 个新连接，并对连接的生存期，最大值等参数进行设置，保证持久层的正常工作。

### 5.3 HAZOP 模块的设计

HAZOP 模块是由一些预先设置好的经验信息，或者是某测量点的预设阈值组成。这里的 HAZOP 模块主要实现的是将从测量点得到的测量数据与预设的值做比较，并且按照一定的风险模型加以评估，若超出了界限或满足了风险发生的条件，则开始报警，或开始通知控制点执行控制指令。

HAZOP 模块包含一些算法，这些算法参照风险预警模型，或是神经网络中的自学习方法，将测量点得到的值和相关的运行参数作为模型的参数值带入，假设这种算法被称为 `f_HAZOP(object parameter1,object parameter2,...,object parameter3)`。HAZOP 模块在数据处理模块每得到一个测量点发回的数据时使用 `f_HAZOP()` 方法进行运算，根据得到的结果处理接下来的行为，程序流程图如下：



### 5.3.1 HAZOP 模块与 DCS 系统的整合

HAZOP 分析模块构成之后，按照系统流程分析，应当加载至监听器每监听到一个数据报文之后。

HAZOP 模块的实例化由数据处理模块决定，也就是说每一个数据处理模块有自己独立的 HAZOP 分析模块，在内存上与数据处理模块共享数据。

如上图所示，当新接收到的数据被归置到正确的内存空间中后，系统流程接着进入 HAZOP 分析模块，按照预设的阈值对刚接收到的数据进行判断，当超过范围，则通知控制点模块干预相关设备，并记录报警日志。



## 5.4 用户控制界面设计

用户界面部分包含整个聚丙烯反应流程图，每个设备上标注出控制点和测量点，用不同的颜色和线条区别管线和用料。

按照目前软件设计的流行趋势，减少降低管理员客户机的复杂程度，使用浏览器当作管理员的界面。传统的图片形式无法满足动态显示的要求，用 APPLANT 会占用大量的资源。在这种情况下，2006 年 W3C 组织推出的 SVG 可以满足要求，并且具有良好的扩展空间。

### 5.4.1 SVG

SVG 严格遵从 XML 语法，并用文本格式的描述性语言来描述图像内容，因此是一种和图像分辨率无关的矢量图形格式。SVG 图形格式具有以下优点：

图像文件可读，易于修改和编辑

与现有技术可以互动融合。例如，SVG 技术本身的动态部分（包括时序控制和动画）就是基于 SMIL 标准。另外，SVG 文件还可嵌入 JavaScript（严格的说应该是 ECMAScript）脚本来控制 SVG 对象。

SVG 图形格式可以方便的建立文字索引，从而实现基于内容的图像搜索。

SVG 图形格式支持多种滤镜和特殊效果，在不改变图像内容的前提下

可以实现位图格式中类似文字阴影的效果。

SVG 图形格式可以用来动态生成图形。例如，可用 SVG 动态生成具有交互功能的地图，嵌入网页中，并显示给终端用户。

SVG 主要支持以下几种显示对象：

1. 矢量显示对象，基本矢量显示对象包括矩形、圆、椭圆、多边形、直线、任意曲线等
2. 嵌入式外部图像，包括 PNG、JPEG、SVG 等
3. 文字对象

系统的界面分为两个部分：带测量点、控制点的整体生产流程图和各测量点的柱状图。结合 SVG 支持的现实对象，可以构建出一个完整的用户界面。

使用 SVG 编辑工具（例如 ADOBE ILLUSTRATOR）绘制聚丙烯生产线流程图，也可用 CAD 工具绘制后转换。

对单个测量点仪表的数据显示绘制柱状图方式，每隔一个时间周期根据该测量点的最新数据绘制柱状图。以下是一个完整的 SVG 文件的示例，用于实现这个柱状图。

每一个 SVG 文件实际上是一个 XML 文件的格式，开头的部分有 XML 的通用标签：

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Generator: Adobe Illustrator 13.0.0, SVG Export Plug-In .
SVG Version: 6.00 Build 14948) -->
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
```

SVG 的正式内容开始和结束的标志是<SVG></SVG>标签, 每个 XML 节点的属性描述了该节点的一些属性值:

```
<svg                                version="1.1"                id="chart1"
xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink" x="0px" y="0px"
    width="80.566px" height="130.478px" viewBox="0 0 80.566
130.478" enable-background="new 0 0 80.566 130.478"
    xml:space="preserve">
```

SVG 中每一个<g></g>标签中表示的是一个图形组成单元, 用于表现出设计者思路中的各图形组成部分之间的层次关系。<Line></Line> 标签描述的是最基本的线段对象, stroke 表示的是线段自身的属性例如颜色, 宽度等, x1,y1,x2,y2 表示的是线段两顶点在坐标系中的位置。在接下来的一段代码中, 绘制了一个带参数的小坐标系, 用于显示某测量点数值的变化。

```
<g>
  <g>
    <line fill="none" stroke="#000000" x1="6.783"
y1="119.239" x2="55.783" y2="119.239"/>
    <g>
      <g>
        <line fill="none" stroke="#000000" x1="6.783"
y1="119.239" x2="6.783" y2="116.764"/>
        <line fill="none" stroke="#000000" x1="55.783"
y1="119.239" x2="55.783" y2="116.764"/>
      </g>
    </g>
  </g>
```

```

    <g>
      <g>
        <text transform="matrix(1 0 0 1 2.572 120.1003)"
font-family="'AdobeSongStd-Light-GBpc-EUC-H'"
font-size="2.2969">100</text>
        <text transform="matrix(1 0 0 1 3.7205 100.3005)"
font-family="'AdobeSongStd-Light-GBpc-EUC-H'"
font-size="2.2969">80</text>
        <text transform="matrix(1 0 0 1 3.7205 80.5007)"
font-family="'AdobeSongStd-Light-GBpc-EUC-H'"
font-size="2.2969">60</text>
        <text transform="matrix(1 0 0 1 3.7205 60.7)"
font-family="'AdobeSongStd-Light-GBpc-EUC-H'"
font-size="2.2969">40</text>
        <text transform="matrix(1 0 0 1 3.7205 40.9001)"
font-family="'AdobeSongStd-Light-GBpc-EUC-H'"
font-size="2.2969">20</text>
        <text transform="matrix(1 0 0 1 4.8689 21.1003)"
font-family="'AdobeSongStd-Light-GBpc-EUC-H'"
font-size="2.2969">0</text>
      </g>
      <line fill="none" stroke="#000000" x1="6.783"
y1="119.239" x2="6.783" y2="20.239"/>
      <g>
        <line fill="none" stroke="#000000" x1="6.783"
y1="119.239" x2="8.008" y2="119.239"/>
        <line fill="none" stroke="#000000" x1="6.783"
y1="99.439" x2="8.008" y2="99.439"/>
        <line fill="none" stroke="#000000" x1="6.783"
y1="79.639" x2="8.008" y2="79.639"/>
        <line fill="none" stroke="#000000" x1="6.783"
y1="59.839" x2="8.008" y2="59.839"/>
        <line fill="none" stroke="#000000" x1="6.783"
y1="40.039" x2="8.008" y2="40.039"/>
        <line fill="none" stroke="#000000" x1="6.783"
y1="20.239" x2="8.008" y2="20.239"/>
      </g>
    </g>
  </g>
</g>

```

<RECT>标签表示的是矩形,  $x, y$  是这个矩形左上方顶点在坐标系中的位置,  $stroke$  是这个矩形的粗度,  $width$  和  $height$  参数分别表示的是这个矩形的宽度和高度。

```
<rect      x="13.643"      y="20.239"      stroke="#000000"
stroke-width="0.5" width="35.28" height="99" id = "bar1"/>
</g>
</svg>
```

<Text>标签标识的是文字类型, 属性值中可以填写如字体, 大小, 颜色等定义值。<Text></Text>之间的是文字的具体内容。

```
<text      transform="matrix(1  0  0  1  6.03  127.8806)"
font-family="'ArialMT'" font-size="8" id="text0">Number</text>
<text      transform="matrix(1  0  0  1  73.616  129.24)"
font-family="'ArialMT'" font-size="8" id="text1">0</text>
</svg>
```

以上代码结合在一起, 绘制出了一个带坐标系柱状图, 下方还有文字显示, 这个图可以用来显示测量到的数据, 模拟传统 DCS 仪表盘的现实效果。

### 5.4.2 Ajax

传统的 WEB 浏览器无法主动的实时反映数据的变化, 除非有用户发起的刷新。为了使界面能更贴近仪表的效果, 必须实现数据的动态刷新。使用 AJAX 框架可以很好的解决这个问题而不去人为的刷新界面。利用 DWR 开发包将界面与数据库联系起来, 动态的将数据和 SVG 结合起来, 实现更好的显示效果。

AJAX 全称为 “Asynchronous JavaScript and XML” (异步

JavaScript 和 XML)，是一种创建交互式网页应用的网页开发技术。

传统的 web 应用允许用户端填写表单 (form)，当送出表单时就向 web 服务器发送一个请求。服务器接收并处理传来的表单，然后送回一个新的网页，但这个做法浪费了许多频宽，因为在前后两个页面中的大部分 HTML 码往往是相同的。由于每次应用的沟通都需要向服务器发送请求，应用的回应时间就依赖于服务器的回应时间。这导致了用户界面的回应比本机应用慢得多。

与此不同，用 AJAX 实现应用可以仅向服务器发送并取回必需的数据，它使用 SOAP 或其它一些基于 XML 的页面服务接口 (界面)，并在客户端采用 JavaScript 处理来自服务器的回应。因为在服务器和浏览器之间交换的数据大量减少 (大约只有原来的 5%)，结果就能看到回应 (实际上就是服务器回应) 更快的应用 (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，这样可以 web 使服务器要消耗处理时间也减少。

使用 Ajax 的最大优点，就是能在不更新整个页面的前提下维护数据。这使得 web 应用程序更为迅捷地回应用户动作，并避免了在网络上发送那些没有改变过的信息。

#### 5.4.3 DWR

DWR(Direct Web Remoting)是一个 WEB 远程调用框架.利用这个

框架可以让 AJAX 开发变得很简单。利用 DWR 可以在客户端利用 JavaScript 直接调用服务端的 Java 方法并返回值给 JavaScript 就好像直接本地客户端调用一样。

首先在系统中，安装 JAVA 传统的方式编写一个类，例如这里在实现 DCS 界面时，需要从数据库中读出某测量点最新的数据。在系统中构建了 ReadDeviceMod 这个模块，其中有一个方法用于从数据库中读出某测量点最新的数据。

使用 DWR 框架时，首先要在 dwr.xml 文件中声明需要发布出来的类和方法：

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dwr PUBLIC "-//GetAhead Limited//DTD Direct Web
Remoting 1.0//EN" "http://www.getahead.ltd.uk/dwr/dwr10.dtd">
<dwr>
  <allow>
    <create creator="new" javascript="ReadDeviceMod">
      <param name="class" value="HAZOP.ReadDeviceMod" />
    </create>
    <create creator="new" javascript="GetMpValue">
      <param name="class" value="HAZOP.GetMpValue" />
    </create>
  </allow>
</dwr>
```

在展示用的 JSP 页面中，DWR 需要加载一下三个文件

```
<script type='text/javascript'
src='/HAZOP2/dwr/util.js'></script>
<script type='text/javascript'
src='/HAZOP2/dwr/interface/ReadDeviceMod.js'></script>
<script type='text/javascript'
src='/HAZOP2/dwr/engine.js'></script>
```

接下来需要使用回调函数执行 JAVA 对象中的方法，将结果传递给 DWR。之后就可以修改 SVG 对象中对应的参数值，例如将 SVG 矩形的高度值修改成通过 DWR 读出的值，用于表示柱状图的高度。

程序中还可以加入判断语句，若读出的数据大于设定的阈值，可以将柱状图的颜色设置为红色；若在范围内则用绿色显示。

以下为实现上述功能的代码：

```
function getValue(){
    ReadDeviceMod.getMpValueFromDb('T1t03',loadinfo);
}

function loadinfo(data){
    DWRUtil.setValue("reply1", data);
    thevalue = data;
    var svgDocument = window.sample.getSVGDocument();
    chart = svgDocument.getElementById("bar1");
    chart.setAttribute("height",thevalue);
    if(thevalue > 80) chart.setAttribute("fill","red");
    else chart.setAttribute("fill","green");
    svgDocument.getElementById("text1").getFirstChild().set
Data(thevalue);
}
```

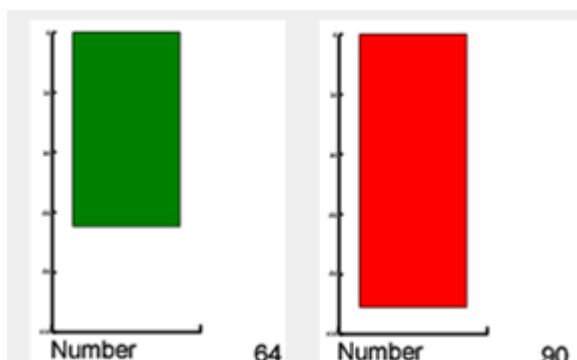
为了实现图标的动态演示，使用函数，每隔特点的时间更新页面，这种更新不同于整个浏览器的刷新，不会占用过多的资源，能实现动画效果。

以下的代码设置每隔 1 秒钟执行一次读取数据库的方法。

```
function run(){
    window.setTimeout("run()", 1000);
    getValue();
}
```



将 SVG 和 AJAX 结合起来，最终可以实现如图所示的效果。



根据自定义的显示风格，颜色等参数，较好的模拟了仪表盘的显示效果。

## 5.5 配置文件设计

从保障统一的系统配置角度来考虑，设计中采用全局性的配置文件对各个模块的初始化数据进行管理，配置文件使用 XML 方式存放配置信息。

统一的配置文件有利于技术人员整体的对生产设备进行定义，能保证程序运行中采用相同的参数，避免错误的 inconsist 的数据，有利与软件的开发和使用。

使用 XML 作为配置文件的载体有以下好处：

1. 可读性强，信息量充实。
2. 模块之间的逻辑关系清晰，便于层次的划分和管理。
3. XML 是通用标准，便于和其他系统集合。

XML (Extensible Markup Language) 即可扩展标记语言，是

SGML(Standard Generalized Markup Language, 标准通用标记语言)。Xml 是 Internet 环境中跨平台的, 依赖于内容的技术, 是当前处理结构化文档信息的有力工具。扩展标记语言 XML 是一种简单的数据存储语言, 使用一系列简单的标记描述数据, 而这些标记可以用方便的方式建立, 虽然 XML 占用的空间比二进制数据要占用更多的空间, 但 XML 极其简单易于掌握和使用。

为了读取测试文件, 还需要编写一个配置文件读取分析模块, 该模块负责读取特定的配置文件参数, 并将参数更新至内存中的各模块对应变量处。

由于面向对象的程序设计中反应装置的定义有很强的逻辑层次性, 所以在配置文件中能够很好的体现出对象的层次, 采用 XML 格式作为配置信息载体尤为合适。

以下是对反应流程中, 某装置的 XML 定义如下:

```
<?xml version="1.0" encoding="UTF-8"?>
<HAZOP>
  <Device DEVICEID="T01">
    <MPList MPLID="T01@1500" DEVICEID="T01" PORT="1500"
IP="210.42.24.87">
      <MeasuringPoint MPID="T1t01" MPTYPE="T"
ThresholdValue="100.0" Porttlistener="1500"
DeviceID="T01"></MeasuringPoint>
      <MeasuringPoint MPID="T1t02" MPTYPE="T"
ThresholdValue="100.0" Porttlistener="1500"
DeviceID="T01"></MeasuringPoint>
      <MeasuringPoint MPID="T1t03" MPTYPE="T"
ThresholdValue="100.0" Porttlistener="1500"
DeviceID="T01"></MeasuringPoint>
    </MPList>
```

```
</Device>  
</HAZOP>
```

在这个配置文件中，父节点是 Device，对应软件系统中的 Device 对象，该节点的属性表明了本设备的 ID 值。Device 对象的子节点为测量点组，在这一层次的属性中，包含了测量点具有的共同属性。第三个层次为测量点，每个测量点节点的属性描述了这个测量点的参数值。

## 5.6 测试平台设计

为了达到系统压力测试，需在正式使用前设计压力测试平台。

测试平台中运行的数据参造武石化聚丙烯生产线的实际数据情况，关键装置，测量点，控制点的编码，位置，数据等均参造实际情况生成。

测试平台中有一个模拟产生测量点数据的多线程客户程序，该模块按照实际数据的情况以恒定的速度产生模拟数据，并将这些数据发送至监听器。为了达到更好的测试效果，该模块还会在特点的时间产生一些异常数据，便于测试 HAZOP 模块分析是否正常。

测试平台的另一个部分是一个自动加载监听器，测量点，控制点等模块的辅助启动模块，这样在测试的时候无需手动启动每个模块，提高了测试的准确性和效率。

测试平台中将运行的数据局存储至对应的文件中，包含每一条测量点得到的数据，HAZOP 分析模块发现的报警信息，系统的启动运行状态数据等。每条记录均包含时间，时间类型，编号，详细描述等内容。

### 5.6.1 测试平台设计出发点

在系统的开发中，测试是一个必不可少的环节，但是由于化工行业的特点，在系统构建阶段无法去实际的生产线上测试。鉴于此考虑，一个模拟聚丙烯生产的测试平台是必须的。该测试平台的设计出发点是围绕两个要点考虑的：一是模拟实际生产中的数据，包括数值，数据量。另一个要点是要能模拟出可能出现的异常情况，用于检测 HAZOP 分析模块是否能发现异常并加以处理。

从软件的性能上来说，测试平台要能为各个模块如监听器，测量点，控制点提供运行的平台环境，要能加载这些对象，保持各个模块的运行能各负其职，平台的运行要有日志记录，为可能出现的异常情况提供分析依据。

平台的一个重要功能是按照实际生产中测量点的类型按照一定的时间周期模拟产生出数据，包括温度，压力，流量等。这些数据要尽可能的贴近实际生产中产生的数据，提高模拟的真实性。越真实的贴近实际生产的数据，越能检测出系统设计中存在的问题。

### 5.6.2 压力测试和异常测试

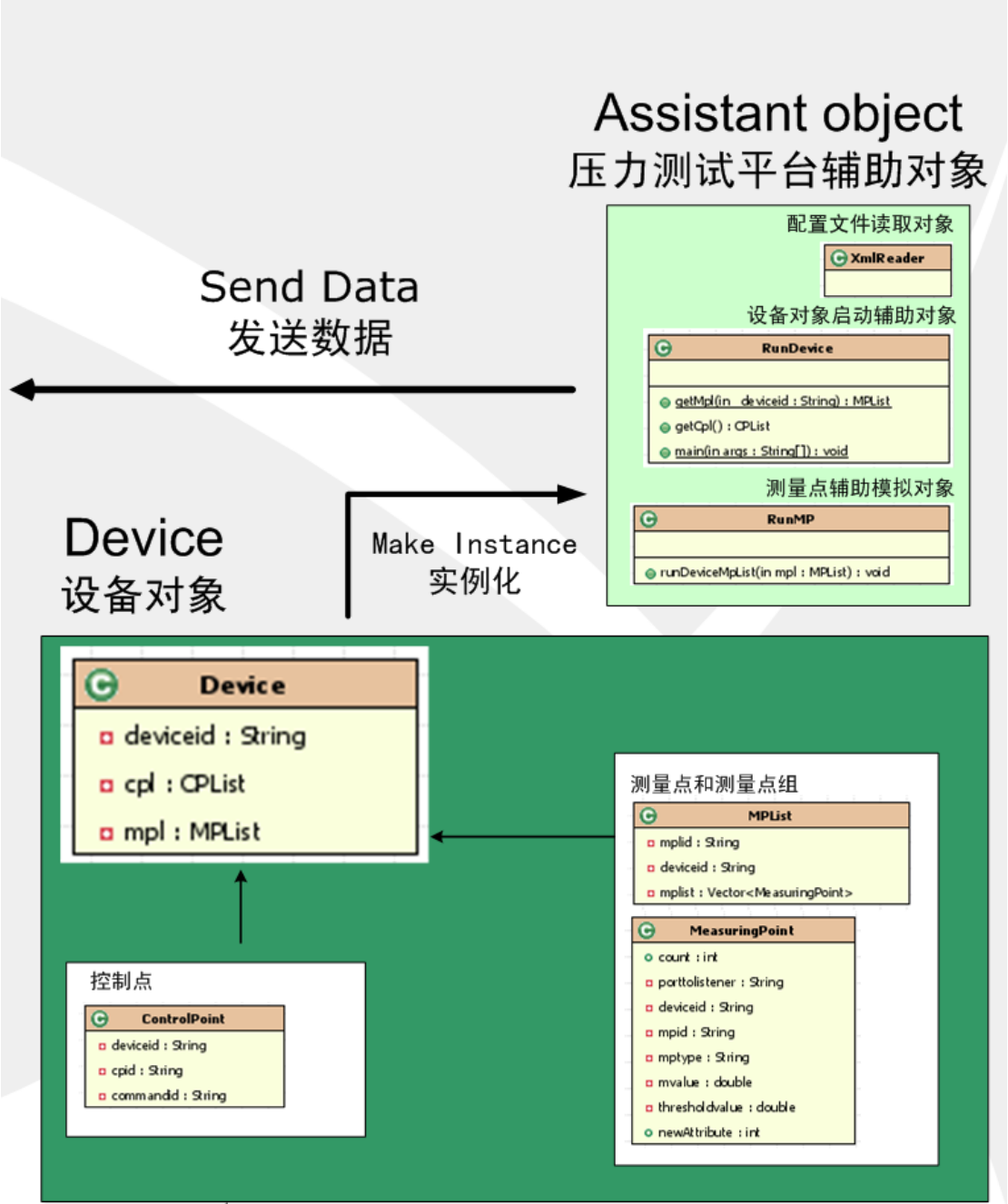
化工生产中的数据变化比较频繁，并且若出现异常的反应后有可能瞬

间出现危险，所以检测数据时要设定适当的检测周期，在平台中要允许设置这些参数。为了能最大限度了解系统所能承受的数据量，应当能模拟出大量的数据，甚至超出预期的情况。例如，实际的生产中，检测的时间间隔可能是 0.5 秒，测试平台可以将测量点对象产生数据的时间周期设置成 0.1 秒甚至更快，以检验软件系统的运行效果。

平台的另一个作用是要模拟出一些异常的数据，这些数据应该是随机产生的，涉及各个检测点。异常数据可以是超过阈值的某个值，也可以是针对某种检测值产生的特定值。异常数据产生的时间应该遵从随机的特点，另一种情况为了测试某种特定的场合也可以按照测试人员的意愿产生。这些特点在测试平台中都能有所表现。

### 5.6.3 测试平台的组成

测试平台主要由一些模块的辅助启动类和配置文件组成。测试应该能全面快捷的进行。由于各模块有一大部分是独立的线程，为了能准确的启动各个模块，需要编写辅助类完成监听器，测量点组，控制点的启动。测试中各个模块的启动需要有一定的顺序，辅助启动类可以保证这些模块的顺序启动和加载。



运行的结果是测试中最需要被关注的内容，所以在测试平台中需要对测试的日志和结果进行详细的记录和保存，例如日志中要记录每次运行的开始时间，参数描述。

下面的结果是系统在测试时的报警日志记录，内容包含了发生的时

间, 设备 ID, 测控点 ID, 测量数据类型和预设的阈值情况。

>Caution: Sun May 31 09:54:31 CST 2009 R201 alert for R201 P 9.0

>Caution: Sun May 31 09:54:32 CST 2009 R201 alert for R201 P 5.0

>Caution: Sun May 31 09:54:33 CST 2009 R201 alert for R201 T 96.0

>Caution: Sun May 31 09:54:34 CST 2009 R201 alert for R201 P 7.0

>Caution: Sun May 31 09:54:35 CST 2009 R201 alert for R201 P 9.0

>Caution: Sun May 31 09:54:36 CST 2009 R201 alert for R201 T 96.0

以下代码展示了如何实例化监听器, 设备对象读取模块和测试平台数据记录模块。

```
Listener rl = new Listener();
rl.setPort("1500");
Thread hll = new Thread(rl);
hll.start();

ReadDeviceMod rd = new ReadDeviceMod();
rd.setMpl(rl.getMpl());
Thread trdl = new Thread(rd);
trdl.start();

BufferDataReader bufferdatareadermod = new
BufferDataReader();

bufferdatareadermod.setBuffData(rl.getBufferdata());
bufferdatareadermod.setFilename("c:\\nlog1.txt");
Thread tbdrmod = new Thread(bufferdatareadermod);
tbdrmod.start();
```

代码中先实例化了一个监听器, 设置 1500 端口为监听端口, 随后实例化了数据读取模块和数据清理模块, 并且指定了日志文件的存放位置。启动了所有的线程 (即 Thread.start()方法的调用)。

## 5.7 多线程协同工作

多线程协同工作是以数据为核心的，系统中的数据对象是多线程处理的对象，各个线程在不同的时间点对各自的目标数据块进行操作。

完整的 DCS 系统由多个模块组成，每个模块各尽其责，但又彼此关联。在软件系统中，每个模块由独立的进程或线程完成。不同的线程可以分布在不同的机站上，在测试环境中也可以部署在一台计算机上。

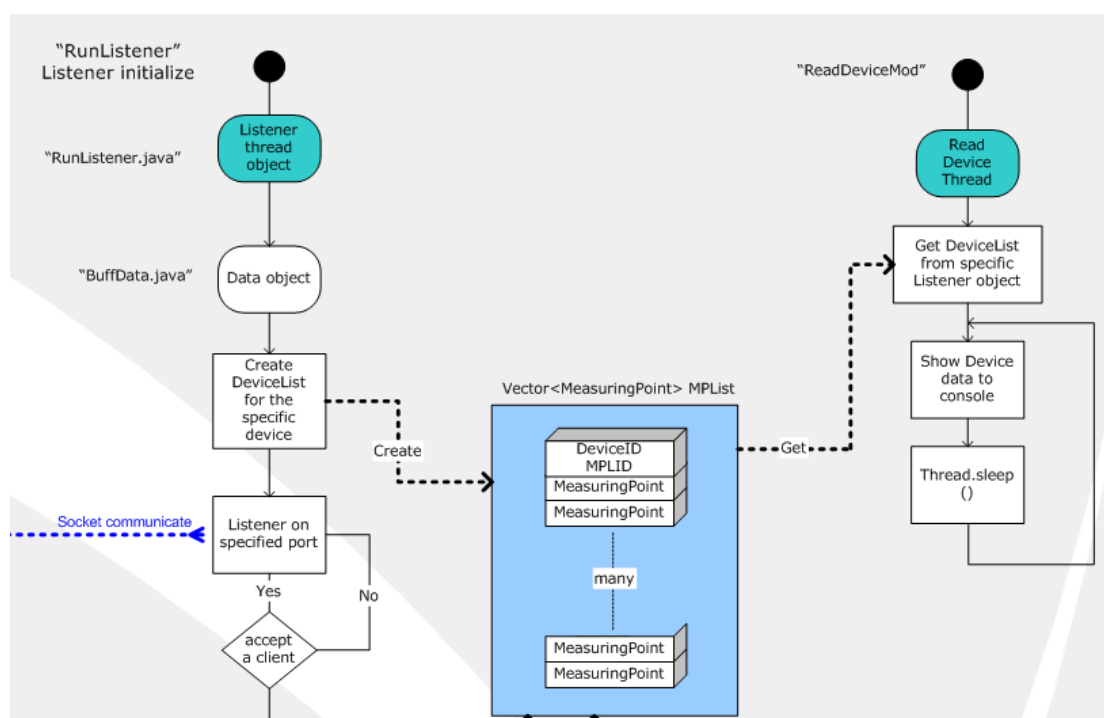
由于系统在运行中，每个进程和线程需要有条不紊的工作，如何保证彼此之前能正常运行是需要在软件设计上重点考虑的。由于本系统关注的重点是数据，模块之间的通信传递的也是数据。按照设计模式中的成熟模型，模块之间采用适配器方式实现，即彼此之间的联系是事务级的。接收到了请求则处理这些内容。

线程之间需要传递的数据，采用共享的数据空间实现。由于在之前的设计中，对数据结构进行了独立的封装，使得数据对象可以被不同的线程使用。

JAVA 中对于共享数据空间的定义比较智能。可以对读取，修改数据的方法加入 `Synchronized` 关键字，做到对某一个数据块读写时保护一致性。声明全局数据变量，使用 `Get` 和 `Set` 的方法，在进程或线程间将数据对象作为参数传入，实现共享数据对象。

下图展示了数据对象是如何在两个线程中被共享的。





例如，listener 类 r1，在实例化 r1 的时候，r1 的构造函数将生成新的内存空间 mp1，用于读取 mp1 数据的线程类 ReadDeviceMod 在实例化的时候，先从 r1 类中 get 出 mp1 对象，再将 mp1 对象 set 到 rd 对象中，这样的做法类似于 C 语言中的指针，由于在 JAVA 中没有指针的定义，采用这种方法即可实现共享内存空间 mp1。

```

Listener r1 = new Listener();
r1.setPort("1500");
Thread h11 = new Thread(r1);
h11.start();
    
```

```

ReadDeviceMod rd = new ReadDeviceMod();
rd.setMpl(r1.getMpl());
Thread trd1 = new Thread(rd);
trd1.start();
    
```

## 5.8 本章小结

本章详细介绍了控制点对象，数据处理模块，数据持久化模块，用户界面模块，配置文件模块，HAZOP 模块的设计思路和实现方法。最后介绍了这些模块是如何共享数据，协同工作的。标准软件的应用为构建一个更为开放、高效的 DCS 系统打下了基础。

## 第六章 结论

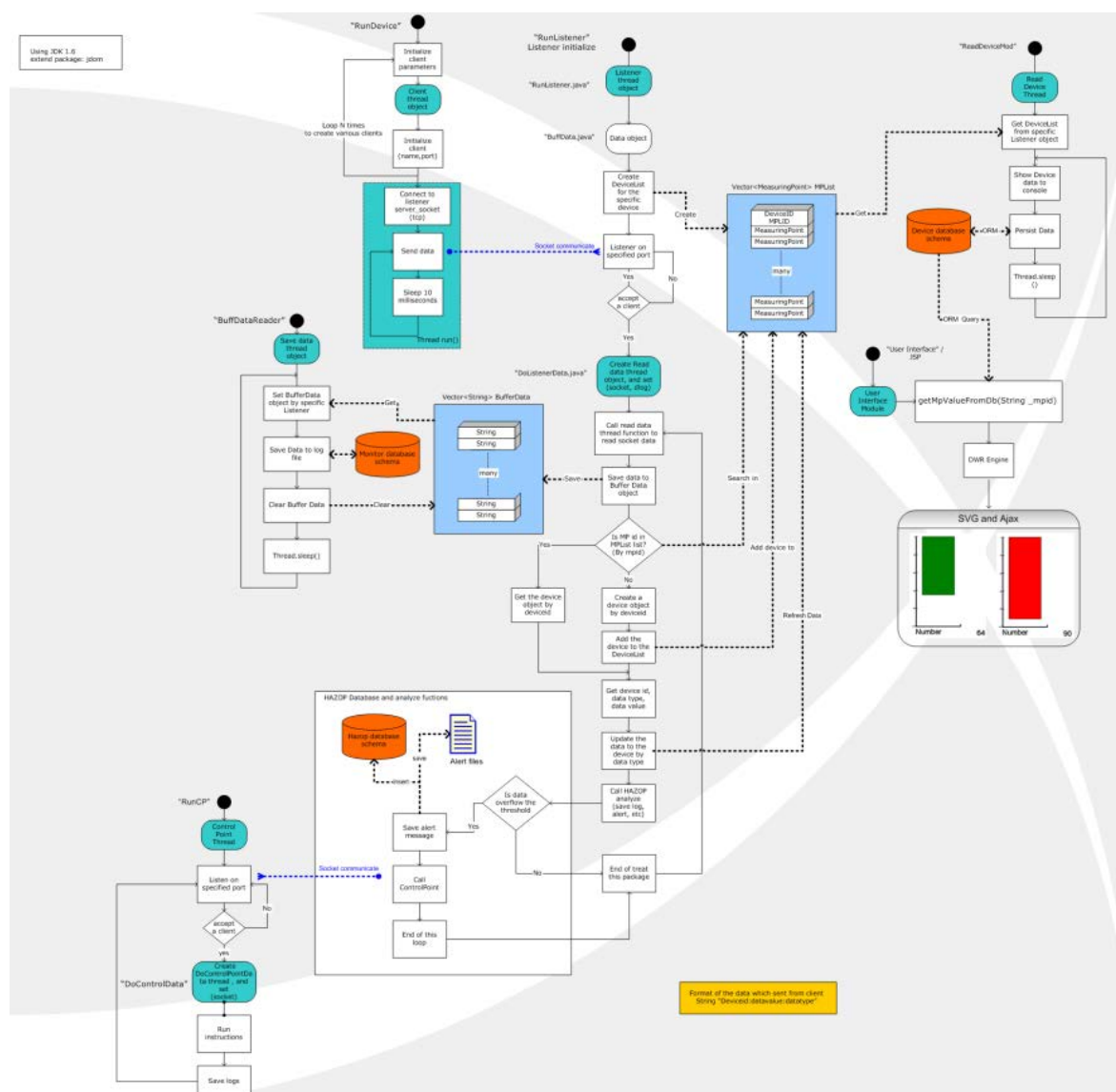
### 6.1 研究结果

系统经过分析，开发，测试，修改，再测试的流程后，形成了一套可以在实验室中运行的模拟 DCS 展示系统。

在研究中，完善了 DCS 模块分析文档，每个功能模块的 JAVA 语言构建和编码。制定了通信的数据结构，实现了模块间的数据通信标准。系统实现数据持久层，使用连接池控制与数据库的连接。界面方面使用 SVG 构建了生产线的工艺流程图，构建了部分模拟仪表显示的 SVG 单元，实现了 SVG 与 AJAX 的结合，从数据库中读取数据加以实时显示。

为了对系统进行测试，构建了压力测试平台，日志记录系统，规定了系统配置文件规则。

整体的程序流程图如下：



## 6.2 测试情况

为了更好的检测运行情况，设计了三种不同的测试方案。第一种测试方案，监听器，测量点，控制点，数据库均部署在不同的测试计算机上。采用 LINUX 操作系统，数据库采用 MYSQL。第二种测试方案，所有的模

块集中在一台测试计算机上，采用 Windows 操作系统，不使用数据库，用文件系统记录结果。第三种方式是混合第一，第二两种方式，监听器在 Windows 操作系统上，测量点对象分布在多台不同操作系统的计算机上。

在这三种测试方案中，测试平台模拟了 4 个设备，共计 40 个测量点，按照每 0.1 秒发送一个测量数据的频率，向监听器发送数据。数据持久化进程每隔 1 秒更新一次测量点的最新数据至内存，并将其持久化。用户界面每隔 1 秒显示该测量点的数据。数据处理模块每 2 秒处理一次数据缓冲区，将部分日志写至文件系统。

平台还对 HAZOP 进行了测试，程序中规定每个一定的周期发送一个会产生异常的数据。在测试的过程中，发现了这些异常数据将立刻通知控制点。整个测试期间所有的异常均能准确快捷的被发现，一旦发现了异常则会立即与控制点对象联系。

测试的数据来自于武石化的实际操作手册，设备的编号，命名，参数的阈值，模拟产生的数据均较为贴近实际情况。

经过近三周的不间断运行，三种方案表现均正常，没有出现死机，内存溢出等错误，所有的日志均有保存，网络通信稳定，界面显示正常，效果较好。以下为监听器，测量点对象，控制点对象在运行中的内存使用情况，在系统运行了 3 个星期后，这些数值没有不断的扩大，保持在一个相对稳定的区间内。

java.exe	root	0	01	25,244 K
java.exe	root	0	00	14,748 K
java.exe	root	0	00	16,028 K

### 6.3 研究解决的问题

第一，研究中，针对武石化现有 DCS 系统的不足，重点从现代软件架构的角度，采用面向对象的分析法对整个 DCS 系统进行了抽象和分析。将系统模块化后，HAZOP 分析模块作为一个独立模块则能较为顺利的融合进 DCS 系统，从而共享、分析来自 DCS 系统的第一手数据，起到及时分析生产情况，降低化学反应生产中可能出现的生产风险的作用。

第二，将具有开放性的标准化软件引入 DCS 系统，构建出了耦合度相对较低的系统。新技术可以缓解旧技术、旧架构带来的瓶颈。采用标准的协议，为今后 DCS 系统与其他系统的对接或联合工作打下了基础。对未来可能到来的各种信息技术融合进行了探索。

### 6.4 研究的展望

在下一步的研究中，将继续完成以下的工作：

1. 完善监听器，测量点对象等模块运行中的通信连接功能，增加冗余通信线路和通信状态保障功能。
2. 构建完整的生产线用户操作界面，设计更多种类的显示内容，进一步实现完整的 DCS 展示功能。
3. 引入武石化生产中实际的设备数据和操作流程到测试平台，做更贴近实际生产情况的测试。

## 6.5 结论

HAZOP 分析法引入聚丙烯生产中,将会降低化工生产中可能出现的风险和灾难。将传统的应急预案和操作手册数字化,逻辑化,交给计算机进行辅助处理,能改善管理水平,提高操作的准确性。

用面向对象的分析方法对 DCS 系统进行分析,细化其功能,抽象出不同的模块。这样的做法能够将 HAZOP 分析法以模块的方式融合进 DCS 系统中。也为今后 DCS 系统自身的不断进化和发展打下了良好的体系基础。

采用稳定的高级语言处理 DCS 系统上位机处理的事务,结合标准的通信协议,提高了 DCS 系统软件运行的稳定度,降低了开发的难度。JAVA 语言的平台无关性为 DCS 系统分布式部署,移植,整合旧系统提供了新的解决思路。

新的软件架构思想能消除系统中潜在的瓶颈,优化系统对于数据的处理速度。SVG 技术能改善传统 DCS 系统的显示功能,遵循标准化技术为未来 DCS 系统的发展提供了保证。

## 参考文献

1. 马沛生主编, 化工热力学, 北京: 化学工业出版社, 2005, 1~100
2. Novenario R C, Caruthers M J, Chao K C, Chain-of-rotators equation of state for polar and non-polar substances and mixtures, Fluid Phase Equilib, 1998, 142:83~100
3. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, "Design Patterns Elements of reusable object-oriented software" Addison-Wesley
4. 孔炜 王文平, "SVG 在发电厂设备在线状态图中的应用" 工业控制计算机 2005 年 18 卷 1 期
5. 杨恒, 徐学洲, "基于 SVG 的 Web 页面动态刷新" 微机发展 Vol15 No.12 Dec. 2005
6. 苟媛媛, 刘志勤, "使用 JSP 动态生成 SVG 统计图" 兵工自动化 2004 年 23 卷第 5 期
7. 魏应彬, 周文山, "由 JSP 动态生成 SVG 图形" 维普资讯
8. 刘信安、陶长安等, "化工过程的计算机图形动态模拟及实现方法", 计算机与应用化学, 1996. 13 卷第 3 期.
9. 皮 振 华 , "AJAX+SVG 实 现 实 时 监 控 图 表 " <http://www.ibm.com/developerworks/cn/opensource/os-ajaxsvg/>
10. Jon Ferraiolo, FUJISAWA Jun, Dean Jackson, "Scalable Vector Graphics (SVG) 1.1 Specification" <http://www.w3.org/TR/SVG11/>
11. Zhangjin, Wu; Xingwen, Huang; Ying, Ding; Rui, Zhou; Qingguo, Zhou Source , "A CGI+AJAX+SVG based monitoring method for distributed and embedded system" Proceedings - 2008 the 1st IEEE International Conference on Ubi-Media Computing and Workshops, U-Media2008, p 144-148, 2008, Proceedings - 2008 the 1st IEEE International Conference on Ubi-Media Computing and Workshops, U-Media2008
12. 徐皑冬, 王宏, 杨志家. 基于以太网的工业控制网络[J]. 信息与控制, 2000 年 4 月, 第 29 卷, 第 2 期:182-186  
贾东耀, 汪仁煌 "工业控制网络结构的发展趋势[J]" 工业仪表与自动化装置, 2002 年, 第 5 期:12-14
13. 曹晓钢, 唐勇, "深入浅出 Hibernate" 电子工业出版社
14. Eric Evans, "Domain Driven Design"
15. Brown, James and Martin, Michael Power Engineering , "Designed-in Safety"; Nov2008, Vol. 112 Issue 11, p156-162, 4p
16. Loss Prevention Bulletin , "HAZOP study for the offshore oil and gas industry", Feb2007, Vol. 193 Issue 1, p31-32, 2p; (AN 24208267)
17. Krishnan, G , "What HAZOP studies cannot do". Unni. Hydrocarbon Processing, Oct2005, Vol. 84 Issue 10, p93-95, 3p; (AN 18630884)
18. Kane, Les; Romanow, Stephany, " New safety tool streamlines HAZOPs". Hydrocarbon Processing, Mar2003, Vol. 82 Issue 3, p30, 2p; (AN 9318341)

19. “HAZOP (Book Review)” Chemical Business, Apr2000, Vol. 14 Issue 4, p57, 2p; (AN 3397095)
20. 王永新, “DCS 数据通讯故障的分析”, 热电技术, Cogeneration Power Technology, 2009 年 01 期
21. 秦利彬, “论 DCS 控制系统可靠性的提高” 中国新技术新产品, China New Technologies and Products, 2009 年 02 期
22. 张维勇, 陆阳, “DCS 网络通讯程序的实现方法” 合肥工业大学学报(自然科学版), JOURNAL OF HEFEI UNIVERSITY OF TECHNOLOGY (NATURAL SCIENCE), 1998 年 04 期
23. 秦猛, “DCS 的可靠性分析” 石油化工自动化, Automation in Petro-Chemical Industry, 2008 年 01 期
24. 王建锋; 罗振新; 薛鹏, “影响 DCS 可靠性的因素分析”, 华东电力, East China Electric Power, 2008 年 04 期
25. 王秋红; 孙旭, “石化装置安全度等级与安全仪表系统的设计”, 石油化工自动化, Automation in Petro-Chemical Industry, 2008 年 01 期
26. 张雪; 王峰; 高金吉, “计算机辅助 HAZOP 技术的研究”, 中国安全科学学报, China Safety Science Journal (CSSJ), 2008 年 02 期
27. Golebiewski, Jan; Galeski, Andrzej, “Thermal stability of nanoclay polypropylene composites by simultaneous DSC and TGA”, Composites Science and Technology, v 67, n 15-16, p 3442-3447, December 2007
28. proxool project, “PROXOOL USER GUIDE”, <http://proxool.sourceforge.net/user.html>
29. 江庆标, “炼油化工装置 DCS 仿真培训系统的开发”, 炼油与化工 2006 年 01 期
30. 钟洋. 化工过程辨识方法的研究 [D], 中国优秀硕士学位论文全文数据库, 2008, (11)
31. 庞海涛. XML 在异构数据交换中的研究与应用 [D], 武汉理工大学, 2006 .
32. Joseph Williams. “The Web Services Debate J2EE VS.NET” [J], Communications of the ACM, 2003, 46, 46 (6) :58-63
33. Svein Johannessen, “Time synchronization in a local area network”, IEEE Control Systems Magazine, 2004, 4, (4) :61~69
34. 张大波, 王光兴, “现场总线体系结构研究及通信栈的设计” [J] 东北大学学报(自然科学版), 1998, (02)
35. 王芳, “中小型集散控制系统工程师站组态软件设计” [D] 河北大学, 2005 .
36. 周述科, 梁坚, “使用 VISUAL C~(++).6.0 实现 DCS 组态软件的设计” [J] 仪器仪表用户, 2002, (01) .
37. 刘艾明, “大型储油罐计算所监控系统的研究与实现” [D] 武汉理工大学, 2005
38. Robert Doverspike and Brian Wilson, “Comparison of capacity efficiency of DCS network restoration routing techniques”, Journal of Network and Systems Management, Volume 2, Number 2
39. Dritan Nace and Jacques Carlier, “Distributed rerouting in DCS mesh



- networks", Lecture Notes in Computer Science, Combinatorics and Computer Science
40. Deo Prakash Vidyarthi, Biplab Kumer Sarker, Anil Kumar Tripathi and Laurence Tianruo Yang, "Load Balancing in DCS", Scheduling in Distributed Computing Systems
  41. J. Labovský, L' . Jelemenský and J. Markoš, "Safety analysis and risk identification for a tubular reactor using the HAZOP methodology", Chemical Papers, Volume 60, Number 6 / 2006 年 12 月
  42. Heeyeop Chae, Yeo Hong Yoon and En Sup Yoon, "Safety analysis using an expert system in chemical processes", Korean Journal of Chemical Engineering, Volume 11, Number 3 / 1994 年 7 月
  43. Stephen A. McCoy, Dingfeng Zhou and Paul W. H. Chung, "State-based modelling in hazard identification", Applied Intelligence, Volume 24, Number 3 / 2006 年 6 月
  44. Janusz Górski, Aleksander Jarzębowicz, Rafał Leszczyna, Jakub Miler and Marcin Olszewski, "Tool support for detecting defects in object-oriented models", Enhanced Methods in Computer Security, Biometric and Artificial Intelligence Systems
  45. Edward R. Sim, Guiseppi Forgionne and Barin Nag, "An Experimental Investigation into the Effectiveness of OOA for Specifying Requirements", Requirements Engineering, Volume 5, Number 4 / 2000 年 12 月
  46. K. Metaxiotis, D. Askounis and J. Psarras, "An Object-Oriented Analysis and Design of a Model for Production Planning and Control in Industry", The International Journal of Advanced Manufacturing Technology, Volume 18, Number 9 / 2001 年 10 月
  47. J. Martin, "Principles of Object-oriented Analysis and Design", Prentice-Hall, Englewood Cliffs, NJ, 1993.
  48. S. Ceri and P. Fraternali, "Designing Database Applications with Objects and Rules - The IDEA Methodology", Addison-Wesley, 1997.
  49. F. Bonfatti, P. D. Monari and P. Paganeli, "Modeling manufacturing resources and activities: an ontology", in J. Winsor and R. Gay (ed.), Proceedings of ICCIM ' 95 International Conference, Computer Integrated Manufacturing, Singapore 1995, pp. 101 - 108, World Scientific, Singapore, 1995.
  50. A. Artiba and S. E. Elmaghraby, "The Planning and Scheduling of Production Systems", Chapman and Hall, 1997.

## 致谢

本文是在导师宣爱国副教授、李晓林副教授的悉心指导和亲切关怀下完成的。借本文完成之际，谨向我尊敬的导师致以最衷心的感谢！感谢宣老师、李老师对我的培养、关怀和精心指导。

感谢本课题组的闫志国老师，袁华老师在课题设计，实施方案设计中所给予的精心指导，他们活跃的学术思想和敏锐的洞察力，时时给我以启发。同时，他们一丝不苟的工作作风和豁达开朗的性格也将使我铭记于心。

特别感谢硕士生何佳、赵世平、苗琪媚等同学，与我共同完成了大量的实验工作，并在实验和数据处理计算中给予我热情的支持和帮助。

感谢远在他乡的宋乃王女士帮助我修正了英文摘要。

感谢我的夫人，在我工作繁忙的时候对我的照顾和鼓励。

最后感谢我的家人在我的求学过程中，对我的深深理解、关心、支持和鼓舞！

谨以此文献给所有关心我的亲人和朋友！