

西北工业大学

硕士学位论文

基于CBR的飞机机载设备故障诊断系统的研究

姓名：王波

申请学位级别：硕士

专业：通信与信息系统

指导教师：宋东

20070301

## 摘要

随着航空技术的发展,现代民用飞机系统的规模和机载设备日趋复杂,传统的故障诊断和维护方法,已不能适应现代民用飞机的发展需求,在保障飞行安全、降低维护成本、提高飞机可用性等方面对故障诊断和维护方式提出了新的挑战。目前,在民用飞机的维护与管理中提出了健康监控与预测、健康管理概念并开展了实时故障诊断技术的研究。

按照目前飞机故障诊断领域的发展趋势,论文研究了基于 CBR 的飞机故障诊断方法和技术,以飞机的 BIT 信息为核心,对故障信息的案例表示、故障案例的检索匹配、故障征兆的权重确定等一系列问题进行了研究,并设计实现了一个故障诊断专家系统。

论文首先针对目前机载系统故障诊断的特点,提出了基于飞机系统的 BIT 信息、飞行参数、FDE 等故障征兆完成飞机故障诊断的 CBR 方法。研究了飞机故障案例的基于框架的表示方法,该方法能够准确的表示一个机载设备历史上发生故障的每次诊断经验。采用了粗糙集的方法对故障案例对应的故障征兆进行简约并计算征兆值对于它所表示的 BIT 信息的重要度。进而利用征兆的重要度提出了一个征兆的检索匹配策略,采用重要度越大的征兆先匹配的原则。历史案例也按照案例历史上被成功匹配的次數作为检索的依据。以征兆重要性倒数的大小作为匹配约束条件,建立了局部匹配的过滤策略。在满足局部匹配约束的条件下,利用最近邻算法计算案例的全局相似度,这种推理策略能够快速准确的得到诊断结果。最后设计实现了一个基于 CBR 的飞机故障诊断专家系统,系统能够完成 CBR 系统流程的基本功能,实现了故障诊断的要求,满足了系统设计的目标。

综上所述,论文在故障征兆重要度的确定、案例的匹配、征兆的匹配问题作了深入的研究,并完成基于 CBR 的飞机故障诊断专家系统的设计与实现,设计实现的专家系统经过验证,能较为准确的进行故障诊断,为后续的相关研究工作打下了良好的基础。

**关键词:** 故障诊断, CBR, 粗糙集, 案例相似匹配, 专家系统

## Abstract

With the development of the aviation technique, the scale of the modern civilian plane and the on-board equipments are more and more complex. So it promotes a challenge that how to carry out the fault diagnosis and maintenance for the plane on the base of assuring the safety, low costs and usability. At present, the domain experts raise a concept about healthy monitoring, forecasting and management. Meanwhile, the work of real-time fault diagnose has been researching in the field of civilian plane.

According to the trend of the airplane fault diagnosis, the diagnostic technique and diagnostic method about fault diagnosis based on CBR are studied. The BIT messages are considered as a kernelled factor, in the base of which the case express, case retrieve, and the right of fault symptom are worked over. And then a fault diagnose expert system is established.

Firstly, according to the characteristic of the fault diagnosis on the on-board equipments, a method that accomplishing fault diagnosis about a plane adopting CBR strategy based on BIT message, flight parameter, FDE and other fault symptoms is brought forward. Secondly, the frame method is applied to represent the history fault cases, which can represent the knowledge exactly. The Rough Sets theory is used to process the fault symptoms to reduce the number and dimension of the symptoms and to calculate the weightiness of every symptom. The weightiness is utilized to decide the sequence of symptom retrieved, and the bigger one is matched first. The times of history matched successfully is used to decide the sequence of history cases retrieved. In the matching process, the reciprocal of symptom weightiness is set as a local restrict condition, and Nearest Neighbor method is set as a whole matching strategy, which can achieve the most referenced results veraciously. Finally, a fault diagnosis expert system based on CBR method is designed and realized. The system is tally with the CBR flow and can realize the requirement of the fault diagnosis and meet the designing target.

In conclusion, symptoms and cases matching, symptoms weightiness ascertained are studied deeply, and then a fault diagnose expert system based on CBR is accomplished and validated. All of the work set up a stable foundation for the coming up research.

**Keywords:** fault diagnose, CBR, Rough Sets, case retrieve, expert system

## 西北工业大学

### 学位论文知识产权声明书

本人完全了解学校有关保护知识产权的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属于西北工业大学。学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。本人允许论文被查阅和借阅。学校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。同时本人保证，毕业后结合学位论文研究课题再撰写的文章一律注明作者单位为西北工业大学。

保密论文待解密后适用本声明。

学位论文作者签名：王波

2007年3月26日

指导教师签名：宋东

2007年3月27日

---

## 西北工业大学

### 学位论文原创性声明

秉承学校严谨的学风和优良的科学道德，本人郑重声明：所呈交的学位论文，是本人在导师的指导下进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容和致谢的地方外，本论文不包含任何其他个人或集体已经公开发表或撰写过的研究成果，不包含本人或其他已申请学位或其他用途使用过的成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式表明。

本人学位论文与资料若有不实，愿意承担一切相关的法律责任。

学位论文作者签名：王波

2007年3月26日

## 第一章 绪论

### 1.1 课题的研究背景及意义

随着航空技术的发展,现代民用飞机系统的规模和机载设备日趋复杂,在保障飞行安全、降低维护成本、提高飞机可用性等方面对故障诊断和维护方式提出了新的挑战,传统的故障诊断和维护方法,已不能适应现代民用飞机的发展需求。

上世纪九十年代初期,国外在民用飞机中发展了实时故障诊断技术,提出了健康预测管理的概念,将飞机在地面进行的事后状态检查、故障诊断工作逐步转移到机上实时进行。通过在飞行数据采集器中增加实时诊断模块实现飞机状态监测,通过地面专家系统的研究,实现了维护支持系统(AMOSS)。

经过 10 余年的发展,在飞机实时故障诊断和维护支持技术方面取得了长足的进步。例如波音 B747 飞机的中央维护计算机系统(CMCS);空客 A300-600 和 A320 的飞机电子中央监控(ECAM)系统。中央维护计算机系统(CMCS)采用综合故障诊断技术,对降低虚警率起到了一定的作用。波音 B777 更是兼有中央维护计算机功能(CMCS)和飞机状态监控功能(ACMS)。目前,波音公司已经将基于数据链的飞机远程监控系统投入应用,并首先应用在波音 747 和 777 机型上,大大提升了飞机安全性、维护效率和经济性。

根据民航有关专家预测,到 2008 年~2009 年新一代民用飞机的维修保障将会发生革命性的变化,其维护费用将只有传统维护方式的一半。其中,高可靠的航空电子设备、实时故障诊断和先进的维护支持技术是主要的贡献因素。该技术能“高效率支持维护”,估计能减少维护人力 20%~40%,减少备件量 50%,提高签派率 25%。这项技术实现了基于状态的维护(即视情维护),从而取代了原来基于事件的维护(即事后维护)或与时间相关的维护(即定期维护)。

目前该领域的技术发展趋势正朝着将机载各系统的 BIT 信息与飞行状态数据信息相结合的方向发展,以实现飞机实时故障诊断与维护支持的综合化、智能化和网络化。

国内在民用飞机实时故障诊断与维护技术方面的研究与实现处于起步阶段。

如新舟—60 飞机通过选装国外机载设备, 尽管在分系统 BIT 功能上达到了国外先进水平, 但还没有自主知识产权的机载综合 BIT 系统。

综上所述, 从国内外在该技术领域的研究现状和发展趋势看, 尽快发展我国自主知识产权的飞机实时故障诊断和维护支持系统, 实现关键技术突破, 对提高我国民用飞机的综合性能水平和国际竞争力是非常必要的, 同时具有显著的社会效益和经济效益。

## 1.2 机载设备故障诊断与专家系统的技术简介

### 1.2.1 机载设备故障诊断的目的和意义

机载设备是指飞机上装备的电子设备, 主要包括雷达系统、无线电系统和导航系统的相关设备等。现代飞机的机载设备是完成飞行任务的关键系统, 而机载设备的性能检测与诊断维修则是飞机地面保障的一个重要方面。所谓机载设备故障诊断, 就是要鉴别机载设备的技术状态是否正常, 如果不正常需要确定故障性质、故障部位及故障起因, 并且提出排除故障相应措施的过程。随着航空技术的发展以及飞机自动化水平的提高, 机载设备变的更加复杂, 不仅同一设备的不同部分之间相互关联、紧密耦合, 而且不同设备之间也存在着紧密的联系; 此外某个系统中出现的某些微小故障若不能及时检测并排除, 就有可能造成整个系统的失败、瘫痪, 甚至导致灾难性后果。因此对机载设备进行故障检测与分离, 可以提高装备的保障性和可维修性, 是提高系统安全性、可靠性的重要方面<sup>[5]</sup>。

### 1.2.2 故障诊断技术的发展与现状

当一个系统的状态偏离了正常状态时, 称系统发生了故障, 此时系统可能完全或部分地失去其功能。故障诊断就是寻找故障原因的过程, 包括状态检测、故障原因分析及趋势预测等内容。故障检测与诊断技术发展至今经历了三个阶段: 第一阶段由于机器设备比较简单, 故障诊断主要依靠专家或维修人员的感觉器官、个人经验及简单仪表就能胜任故障的诊断与排除工作; 第二阶段是以传感器技术、动态测技术为手段, 以信号分析和建模处理为基础的现代诊断技术, 在工程中已

得到了广泛的应用。近年来,由于机器设备日趋复杂、智能化及光机电一体化,传统的诊断技术已经很难适应,随着计算机技术、智能信息处理技术的发展,诊断技术进入了它的第三个发展阶段——信号分析、建模与知识处理相融合的智能诊断技术阶段<sup>[5]</sup>。

概括来说,现有的故障检测与诊断方法(Fault Detection and Diagnosis-FDD),可以分为以下四大类:①基于状态估计的FDD方法;②基于参数估计的FDD方法;③基于信号处理的FDD方法;④基于知识的FDD方法。①、②又统称为基于解析模型的诊断方法,核心思想是用解析冗余取代硬件冗余,通过构造观测器估计出系统的输出值,将其同输出测量值相比较,从中获取故障信息。该方法获得了深入的研究,但在工程实践中,由于获得系统精确模型的困难性,限制了其使用范围和效果;基于信号处理的诊断方法,利用信号模型,如相关函数、频谱等,提取诸如方差、幅值、频率等特征值,检测出故障,随着小波变换、分形等分析技术的不断引入,研究和应用领域在迅猛地拓展;基于知识的诊断方法不需要对象的精确数学模型,诊断对象作为一个有机整体被研究,以知识处理技术为基础,诊断问题的求解致力于通过模拟领域专家在推理过程中控制和运用各种诊断知识的行为而获得解决,目前研究工作发展迅速、成果迭出。

### 1.2.3 专家系统与故障诊断

故障诊断专家系统是将人类在故障诊断方面的多位专家具有的知识、经验、推理、技能综合后编制成的大型计算机程序,它利用计算机系统帮助人们分析解决只能用语言述、思维推理的复杂问题,扩展计算机系统原有的工作范围,使计算机系统有了思维能力,能够与决策者进行“对话”,并应用推理方式提供决策建议。专家系统在故障诊断领域的应用非常广泛,长期以来,在航空、航天、电力、机械、化工、船舶等许多领域,故障检测与诊断技术与专家系统相结合,使工程的安全性及可靠性得到保证。故障诊断专家系统除了具备专家系统的一般结构外,还具有自己的特殊性。它具有如下特点<sup>[1]</sup>:

- (1) 知识可以从类似机器和工作实际、诊断实例中获取,即知识来源比较规范。
- (2) 诊断对象多为复杂的、大型的动态系统,这种系统的大部分故障是随机的,



普通人很难判断,这时就需要通过讨论或请专家来进行诊断。但对于一些新型机器,可能无处获得诊断知识,或者对于非定型生产的机器,由于其工作特性和常用机器相比差异很大,知识获取也十分困难。

故障诊断专家系统的发展起始于20世纪70年代末,虽然时间不长,但在电路与数字电子设备、机电设备等各个领域已取得了令人瞩目的成就,已成为当今世界研究的热点之一。在专家系统已有较深厚基础的国家中,机械、电子设备的故障诊断专家系统已基本完成了研究和试验的阶段,开始进入广泛应用。在电路和数字电子设备方面,MIT研制用于模拟电路操作并演绎出故障可能原因的EL系统;美国海军人工智能中心开发了用于诊断电子设备故障的IN-ATE系统;波音航空公司研制了诊断微波模拟接口MSI的IMA系统;意大利米兰工业大学研制用于汽车启动器电路故障诊断的系统。国内在故障诊断专家系统方面起步稍晚,20世纪80年代开始,国内部分高校和科研机构首先在汽车故障诊断领域对专家系统进行了研究,发表了一系列的研究文献,部分文献已经达到国外同等水平,随后在其他电子设备领域和电力应用领域相继进行了故障诊断专家系统的研究,并取得了一定的研究成果,有一些系统已投入了实际运行。具有代表性的专家系统有华中科技大学研制的用于汽轮机组工况监测和故障诊断的智能系统DEST;哈尔滨工业大学和上海发电设备成套设计研究所联合研制的汽轮发电机组故障诊断专家系统MMMD-2;清华大学研制的用于锅炉设备故障诊断的专家系统等等<sup>[1]</sup>。都能够比较成功地进行复杂系统的故障诊断,这表明我国的故障诊断已经进入专家系统故障技术时代。

目前已研究的故障诊断专家系统模型有<sup>[1]</sup>:基于规则的诊断专家系统、基于实例的诊断专家系统、基于行为的诊断专家系统、基于模糊逻辑的诊断专家系统和基于人工神经网络的诊断专家系统。

### 1.3 论文研究内容及组织结构

本文探讨了基于事例推理的基本过程,并结合基于飞机机载设备BIT信息的故障诊断专家系统对推理过程的各个核心环节进行了讨论。在事例表示中采用了基于框架的表示方法,有效的解决了知识表达的可拓性问题。在事例检索中,采用了最近相邻算法和属性过滤法相结合的策略,其中在属性过滤法中采用了粗糙集

的理论对数据进行处理,保证了数据在客观上的真实性。整个方法的引进确保了案例的检索效率和检索的准确性;在案例的学习过程中,总结了三种不满足条件的情况下进行的学习过程,这样可以不断增加事例库的内容,从而提高案例匹配的成功率和准确性。

本文的组织结构如下:

第一章:绪论,简要介绍了故障诊断与专家系统的相关内容,机载设备故障诊断现状。

第二章:介绍了基于 CBR 的专家系统的机理。

第三章:基于 CBR 的故障诊断专家系统的设计与建模,介绍了设计过程中用到的算法以及理论。

第四章:故障诊断专家系统的设计实现,详细讲述了数据库的设计以及专家系统的各功能模块的功能和具体实现方法。

第五章:通过实际运行所设计的专家系统,进行系统总体性能的检验与评估。

最后对所研究的内容进行了总结与展望,提出了系统应该继续完善或是改进的地方。

## 第二章 基于 CBR 的专家系统机理

基于案例的推理 (Case-Based Reasoning, 简称 CBR) 源于认知科学中记忆在人类推理活动中所扮演的角色, 1977 年 Schank 和 Abelson 在这方面的工作被认为是 CBR 的起源。随后美国耶鲁大学 Roger Schank 又探索了记忆中的案例问题解决和学习中的作用, 于 1982 年在《Dynamic Memory》一书中提出了 CBR 认知模型。他的早期思想被其耶鲁大学的学生推广, 1983 年 Janet Kolodner 开发的第一个实际 CBR 应用系统 CYRUS 在计算机上首次实现了 Roger Schank 著作中的许多原理。尔后, 耶鲁大学、佐治亚工学院、马萨诸塞大学又建造了一些系统, 这些系统在法律、烹调、医药、故障诊断等领域证明了 CBR 的有效性和实用性。在 20 世纪 80 年代后期, 美国的 DARPA 项目导致了 ReMind 系统的产生。这被认为是 CBR 从认知科学向人工智能方向转变的标志。如今, CBR 已作为人工智能的一种推理技术。有很多的学者和学术团体在致力于 CBR 的理论和方法研究, 如 Roger Schank, Janet Kolodner, David W Aha, Barry Smyth, Michael M. Richer, David Wilson 等学者和德国政府资助的 Kaiserslautern 大学人工智能和知识系统研究小组、美国海军人工智能实验室的智能决策辅助研究小组等团体。由于 CBR 自身的特点, 更多是应用研究, 并得到商业化。随着对 CBR 研究的深入和推广, 1995 年 10 月在葡萄牙召开了第一届 CBR 国际会议。近年来, 已经开发出了许多 CBR 的专家系统<sup>[6]</sup>。

### 2.1 专家系统概述

专家系统 (expert system) 是人工智能应用研究最活跃和最广泛的领域之一。经过 20 年的研究发展, 到 20 世纪 80 年代中期, 各种专家系统已遍布各个专业领域, 取得很大的成功。我们可以这样定义专家系统, 它是一种设计用来对人类专家的问题求解能力建模的计算机程序。它是一个智能计算机程序系统, 其内部含有大量的某个领域专家水平的知识与经验, 能够利用人类专家的知识和解决问题的方法来处理该领域问题<sup>[10]</sup>。

专家系统按其问题求解的性质可分为解释、预测、诊断、设计、规划、监视、控制、调试、教学、修理这十种类型<sup>[11]</sup>。

### 2.1.1 专家系统的结构

专家系统的结构是指专家系统各组成部分的构造方法和组织形式。系统结构的选择恰当与否，是与专家系统的适用性和有效性密切相关的<sup>[11]</sup>。图 2-1 是专家系统简化的结构图<sup>[11]</sup>。

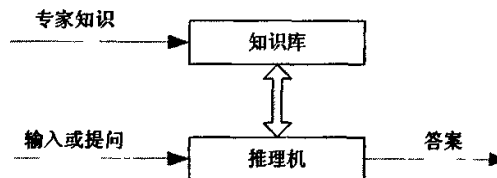


图 2-1 专家系统简化结构图

另外普遍的专家系统还包括人机接口、知识获取机构、解释器这三部分。各部分介绍如下<sup>[11]</sup>：

(1) 知识库：主要包括事实和规则表述的领域知识。知识可来自领域专家、书面记录等多种方面。

(2) 推理机：对推理过程进行控制的模块，它将具体实施某种控制策略，求得问题的答案或证明某个假设的正确性。

(3) 解释子系统：解释、翻译或说明专家系统和用户之间交互作用所涉及的问题、结论、推理过程、因果关系等等。

(4) 知识获取子系统：人类专家或知识工程师将通过这个子系统增加、删除或更改知识库的内容。

(5) 人机接口：直接面向用户，是专家系统与用户交互作用和对话的窗口。在不同的系统中，由于硬件、软件的环境不同，接口的形式与功能有较大的差别。如有的系统可用简单的自然语言与系统交换，而有的系统只能用最基本的方式（如编辑软件）实现与系统的信息交流。

其中知识库和推理机是核心。它们组成的专家系统的一般体系结构如图 2-2 所示。图中，用户是专家系统的使用者，专家和知识工程师是指专家系统的开发者。工作区是指专家系统运行的数据内存区。

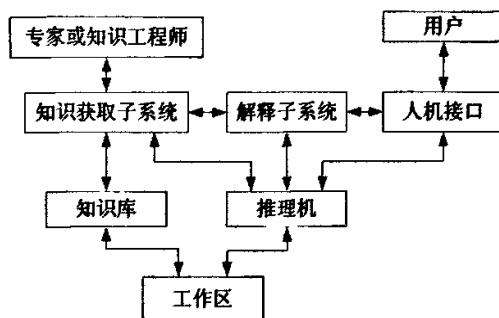


图 2-2 专家系统的一般体系结构

## 2.2 知识库的构建

知识库用于存储某领域专家系统的专门知识，包括事实、可行操作与规则等。为了建立知识库，要解决知识获取和知识表示问题。知识获取涉及知识工程师（knowledge engineer）如何从专家那里获得专门知识的问题；知识表示则要解决如何用计算机能够理解的形式表达和存储知识的问题。

### 2.2.1 知识获取

知识获取的目标在于掌握可指引开发的有关问题知识。这些知识提供问题的见解，为专家系统设计提供材料。它是取得、组织和学习知识的过程。这一过程是开发专家系统的瓶颈问题和最大挑战。

专家系统开发本质上是一种探索性努力。设计者对问题常缺乏初步了解，这样他们不得不对项目采取谨慎的方法。首先寻求对问题的一般解释，并使用这些信息作为探索附加信息的向导。通过这些知识收集和分析的反复过程，逐渐获得对问题的理解和求解此问题的见解。可以用图 2-3 来表示这个循环过程。



图 2-3 知识获取的循环过程

专家系统开发没有终止条件，专家系统可以通过获取更多知识继续提高其性能。

### 2.2.2 知识表示

知识表示是一种用来在专家系统的知识库中对知识编码的方法。它是一种数据结构与控制结构的统一体，既考虑知识的存储，又考虑知识的使用。知识表示实际上就是对人类知识的一种描述，以把人类知识表示成计算机能够处理的数据结构，对知识进行表示的过程就是把知识编码成某种数据结构的过程。

目前常用的知识表示方法如下<sup>[10]</sup>：

(1) 一阶谓词逻辑表示法：一种重要的知识表示方法，它以树立逻辑为基础，是到目前为止能够表达人类思维活动规律的一种最精确的形式语言。它与人类的自然语言比较接近，又可方便地存储到计算机中去，并被计算机作精确处理。

(2) 产生式表示法：一种比较好的表示法，容易用来描述事实、规则以及它们的不确定性度量，目前应用较为广泛。它适合于表示事实性知识和规则性知识。在表示事实性知识和规则性知识时又可以根据知识是确定性的还是不确定性的分别进行表示。

(3) 语义网络表示法：是为了表示概念、事物、属性、情况、动作、事件、规则以及它们之间的语义关系。概念、事物、属性实际上是一种事实性的知识，情况、动作和事件等是一种控制性知识。

(4) 框架表示法：以框架为理论基础发展起来的一种适应性强、概括性高、结构化良好、推理方式灵活、又能把陈述性知识与过程性知识相结合的知识表达方法。

## 2.3 推理机的构建

专家系统使用推理技术对人类的推理过程进行建模。它是专家系统从已知信息获取新的信息的过程。专家系统使用推理机模块进行推理，它结合工作内存中的事实和知识库中的知识，通过这种行为它能够推导出新的信息，并将它加入到工作内存中。图 2-4 显示了这个过程。

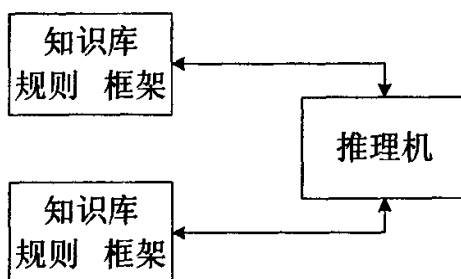


图 2-4 专家系统的推理过程

## 2.4 CBR 的基本概念

基于案例推理 (Case-based Reasoning, CBR) 是由目标案例的提示而得到历史记忆中的源案例，并由源案例来指导目标案例求解的一种策略。它是一种重要的机器学习方法。CBR 是区别于基于规则推理的一种推理和学习模式，它是指借用旧的案例或经验来解决问题、评价解决方案、解释异常情况或理解新情况。CBR 兴起的主要原因是传统的基于规则的推理 (Rule Based Reasoning, RBR) 系统在知识的获取问题上存在困难，不能做到事例的例外处理，整体性能十分脆弱。CBR 克服了 RBR 的众多缺点<sup>[6]</sup>：

(1) CBR 降低了知识获取的难度，不需要得出像规则那样准确和抽象的知识，而是直接使用隐含着难以提取规则的知识片段 (案例)；

(2) 在没有模型存在的情况下，CBR 系统也能建立，而 RBR 系统必须建立应用空间模型；

(3) 系统维护将变得十分简单；

- (4) CBR 系统能以获取新的案例的方法实现自学习;
- (5) 不仅可以学习正面的经验还可以学习反面的经验。

基于案例推理来自认知科学中记忆在人们预期和决策时所扮演的角色, 知识源不是规则而是发生的事例。Riesbeck 和 Schank 对 CBR 的定义是通过访问案例库中的同类案例(源案例)的求解从而获得当前问题(目标案例)解决方法的一种推理技术。在 CBR 中, 知识单元是案例(Case)。按 Roger Schank 的定义, “案例就是指一次经历。实质上, 案例是一组带有相关值的特征, 这些特征描述了一个问题及其结论”。可以看出, CBR 是基于以下两条原则: 现实世界是有规律的——相似的问题有相似的解决方法; 同类的问题会再发生。尽管 CBR 有认知科学为基础, 以人工智能等技术予以实现, 但 CBR 仍然在理论方面很缺乏。由于人们的实践经验, 没有人会否定 CBR 的确有用, 但很难在理论上有严格的证明。这也因此导致了 CBR 中的一些难点的难以解决, 如事例的改写<sup>[6]</sup>。

## 2.5 CBR 的工作机制

CBR 是人们求解现实问题过程的一个合理描述和一种基本方法, 是通过一些基本步骤来实现其思想的方法, 它的工作流程如图 2-5 所示。

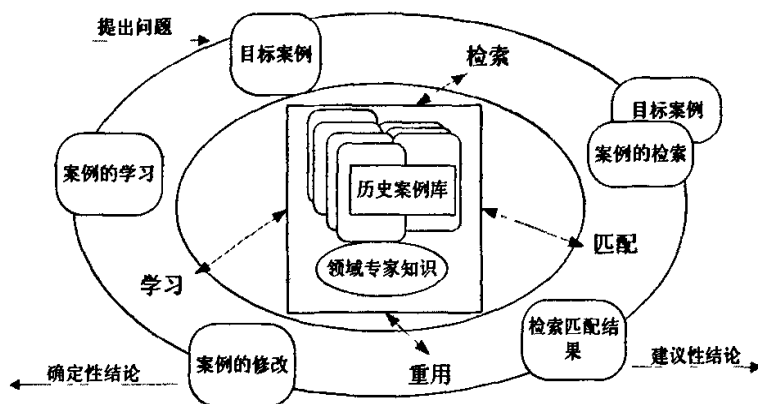


图 2-5 CBR 系统工作流程

CBR 循环可以归纳为如下几个方面: 检索(Retrieve)、重用(Reuse)、修正(Revise)、评价(Review)、保留(Return)。当一个新的待解决问题提出后, 可



以把它抽象为一个新案例，为了解决这个问题我们从案例库中找出一个和新案例最相近的案例，其解决方案就可作为新问题的参考，如果对此方案有什么不满意，可进行修改，修改后的案例成为一个新学习到的案例被存到案例库中，如果下一次碰到类似的问题就可以以此作为参考。从上面可以看到 CBR 的原理和人们日常解决问题的方法基本相同，当我们碰到一个不熟悉的新问题时，我们通常会回忆起以往的成功经验，并以之为参考做出新的解决方案，如果方案成功，则这次经验被作为成功经验记下来，若失败，则作为失败经验记下来，这就是我们不断学习的过程。这些 CBR 的基本步骤是从使用者的角度来考虑的。最简单的实际系统只有案例检索这一项功能，而把其他如案例的改写与是否保存等工作留给使用者。

对 CBR 的研究是从系统的设计者和使用者两方面来考虑，除了要保证实现 CBR 的工作原理外，还需对系统的构造、维护和使用等方面进行研究。CBR 只有一个通用的求解步骤，而少有具体的专用的技术，正是在这个意义上说，CBR 是一种方法而非技术。一方面是 CBR 的发展还不成熟的表现；另一方面使得 CBR 目前是一个开放的系统，这有利于 CBR 本身的发展。当然，CBR 经过 10 年左右的发展也有一些较为通用的技术，如在检索事例的方法中的最近相邻策略和知识引导策略等。由于 CBR 是一种方法而非技术，所以在基于案例推理方面的研究中应用研究较多。

### 2.5.1 案例的表示与存储

案例表示的目的在于把领域专家的知识以一定的表达方法表示出来。一种有效的表示方法应具有简洁性、灵活性和透明性，并且易于诊断推理。表示方法目前已有多种形式，如基于谓词逻辑的形式、基于语义网络的形式、基于规则的形式、基于框架的形式、基于剧本的形式和基于过程的形式等。

案例的存储形式是影响 CBR 系统效率的关键。案例组织的合理，案例检索系统就能够迅速的从案例库中检索出所要的案例，从而使效率提高。

案例中包含的许多信息对以后的推理十分有用，例如结果中既可以包含执行解方案后的结果，还可以包含对这种结果的解释及修正解的方法等，这些信息可以为以后修正失败解做出指导。

## 2.5.2 案例的检索与匹配

CBR 系统的强大功能来源于它能从其记忆库中迅速准确地检索出相关案例。案例索引过程的目标是搞清在将来相似的情形下,在什么时候应该检索一个案例。案例记忆库和检索过程的目的是建设一个结构或过程来得到最适当的案例。

案例检索应达到以下两个目标:

- (1) 检索出来的案例应该尽可能少;
- (2) 检索出来的案例应尽可能与当前案例相关或相似。

具体的检索案例方法有以下四种:

- (1) 模板检索:类似于关系数据库的 SQL 查询,它根据用户输入的问题描述,形成一个检索模板,再依据这个模板在案例库查找与之完全匹配的案例,若有,则返回,否则什么也不返回。
- (2) 分层检索:常与归纳索引相配合使用。其检索过程以归纳索引形成的决策树为基础,由树根开始逐层下降,直到不能下降为止,此时,返回停止点以下所有的案例集。
- (3) 关联检索:又称近邻检索,与近邻索引配合使用。其检索过程是基于近邻索引重定义的案例特征矢量的类似度,将用户输入的新案例与库中案例进行比较,选出相似度高的案例返回给用户。
- (4) 基于知识的检索:由于基于知识的索引没有特定的模式,因此,不同的知识索引及其检索机制差异较大。如果领域的知识以规则的形式表示,则基于知识的检索依赖于一个推理算法。有的将领域的知识反映到特征属性的权重上,并且可以动态的调整权重,使检索更具有灵活性和动态性。

最近邻法是指用户利用与记忆库案例相匹配的输入案例的特征权数和来检索案例。如果检索目标未能很好定义或可用案例不多,这一方法是较好的。单独使用这一方法的最大问题是不可能覆盖全局特征的权值集,在所有情况下难以准确检索到案例。许多问题的特征权重是相互依赖的,在确定要检索的适当案例时,一个给定特征的重要性取决于这个案例的其它特征的权值。

如果检索的目标或案例的结果是被定义的,并且有足够的案例进行归纳比较,那么归纳索引法比最近邻法好。前者的任务是在需要被分类的各种各样的案例之

中,归纳确定哪一特征判别能力最好。归纳法有两个优点:(1)能自动的、客观的、严格的分析案例,确定能区别这些案例的最佳特征;(2)案例可以组织成分层结构供检索用,其检索时间成对数而不是线性增加。归纳索引法的主要缺点是:当使用的案例库包括数以千计的案例时,为了完成归纳,系统需要相当数量的案例来生成判别特征,归纳分析时间长。

基于知识的索引法尝试利用现存的有关案例库案例的知识来确定检索案例时哪些特征是重要的。如果这样的解释性知识是可用的,并且可表示的话,这种方法是可取的。问题是,常常难以代码化足够的解释性知识,在大范围的可能案例输入上完成完备的基于知识索引。因此,许多系统使用与其他索引技术相结合的基于知识的索引法。

### 2.5.3 案例的重用修改

案例的重用被公认为是 CBR 中最具挑战性的问题,也是继案例检索之后的又一个热点。也称为案例的修改。由于案例的重用实质上是用旧的经验解决新的问题。那么显而易见,不管实际经验多么丰富都可能无法解决当前出现的问题。在具体应用技术中,重用大多采用基于规则的方式来改写案例以适应新的情况。因为采用 CBR 的一个主要原因是缺乏领域的知识,通过 CBR 可以绕开知识获取的瓶颈,但采用基于规则的改写方法又会面临知识获取(规则的提取)的问题,又回到了老问题上,因此重用的方法一直没有像检索方法那样有相对成型的方法。更深层的原因是 CBR 在理论方面比较缺乏,尽管都知道 CBR 有效,但很难从理论上给予严格的证明,加上现实世界的多样性,从而导致了案例的重用只能具体问题具体分析而没有相对通用的方法。因此,案例的重用是实现 CBR 目前最困难的问题,不是难在无法构造一种重用的方法,而是构造不出相对通用的重用方法。也因此在这个问题上分为两派:一派主张把重用的工作留给使用者自己来完成,实际上回避了这个问题;另一派则主张要实现重用的功能,但目前也没有一个较通用的技术。这两派并没有根本的分歧,只是在实际应用中对案例重用采取不同的态度。一般来说,在 CBR 中有两种案例修改的方法:结构修改,就是直接应用规则或公式修改所存储案例的结论以适应新的问题;诱导修改,重用得出以前案例结果的规则或公式。采用这种方法时,需另外存储如何得出案例结论的步骤和

知识，以便改写时应用。

具体做法有：

- (1) 不修改：这种策略适用于有复杂原因，但有简单结论的案例，如在银行贷款中的应用；
- (2) 参数调整：属于结构修改，比较源案例和目标案例的特定的参数，在确定的调整方向上进行调整，实际上应用了领域中的一些规则；
- (3) 重实例化：把旧问题的特征用新问题的特征重实例化；
- (4) 诱导重放：检索解决源问题的方法和步骤，在新的问题中予以应用；
- (5) 模型引导：采用因果关系模型来引导修改。

案例修改在很多方面是 CBR 唯一致命的弱点。尽管在很多场合修改是有用的，但决不是必不可少的。事实上，人只凭经验就可以解决所出现的新问题也是不可能的。即使 CBR 系统能主动地思考，也很难只凭案例库中的案例解决所出现的新问题。在可以预见的将来，这个工作只能针对具体问题，具体分析来采用一些重用的方法。根据所在领域的具体规律，构造一种较为通用的重用方法。

#### 2.5.4 案例的学习

案例的学习过程就是采取一定的策略将案例加入到事例库的过程。这种增量式的学习过程使 CBR 系统的知识不断增加，解题能力不断增强，对于 CBR 方法的发展具有积极意义。

除案例库之外，CBR 系统还依赖于多种知识源，包括事例改写知识、索引机制、相似性评估标准等。这些知识的获取对 CBR 系统的构建来说是比较大的负担。因此，一个解决办法就是通过学习来获取这些知识，而 CBR 方法正是一种可行的学习方法。

CBR 系统能方便的学习到新的知识，这是其非常突出的优势，当然，这种优势也会受到案例库容量、系统对推理效率和复杂性要求的影响，许多研究工作表明，如果案例库过大，系统匹配的准确性和效率都会降低，因此，必须对案例库进行增补、删除、修改等操作，即及时维护案例库。

通过案例的表示、存储、检索、重用、学习后，就完成了—次 CBR 的推理过程，图 2-6 为典型 CBR 推理流程，具体过程可能根据不同的要求还会有增减。

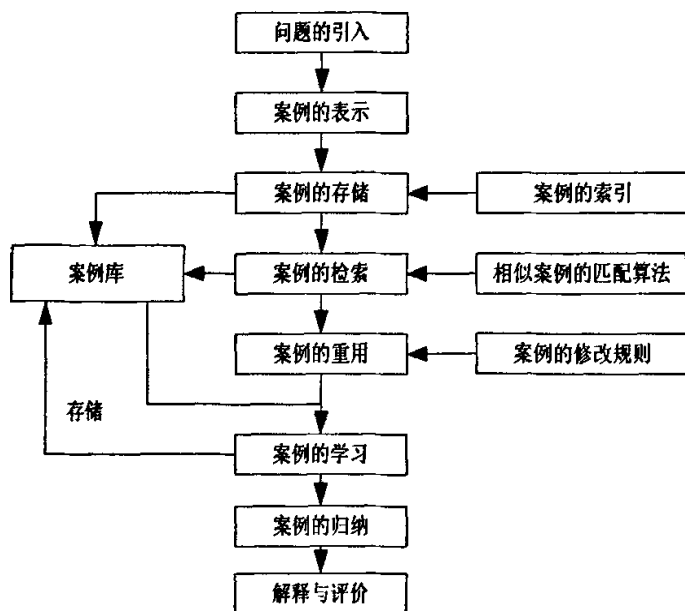


图 2-6 CBR 推理基本流程图

### 2.5.5 案例的评价

评估任务需要在现实环境中应用该实例解答的结果，可以通过询问专家或是现实世界中具体执行任务来实现。这通常是 CBR 系统外部的一个步骤。

## 2.6 本章小节

本章分别对专家系统和案例推理（CBR）系统进行了论述。探讨了专家系统的基本结构，论述了知识库、推理机的构建方法，研究了 CBR 系统的基本概念、工作流程。

### 第三章 基于 CBR 的故障诊断专家系统总体设计

#### 3.1 系统设计目标

本系统作为飞机实时故障诊断系统（如图 3-1）中的地面故障诊断子系统，配合机上故障诊断子系统协同工作。当各机载分系统的故障信息不准确，在机上无法及时判明故障原因时，可以利用飞机的数据传输系统接口，实时将飞行参数和故障信息传回到地面子系统。地面子系统将实时查询飞机电子维护手册和更详尽的故障模式库，并根据飞机历史故障数据库，采用混合智能故障诊断技术和专家诊断系统，对故障原因进行定位，该系统还可以通过地空数据链上传至飞机，实现隔离故障到 LRU，对空、地勤人员应急处置和维护提出建议。同时还可将故障的诊断结果分发到地面上其他相关部门，做好飞机排故的准备，例如准备航材备件，准备维修检测工具等。

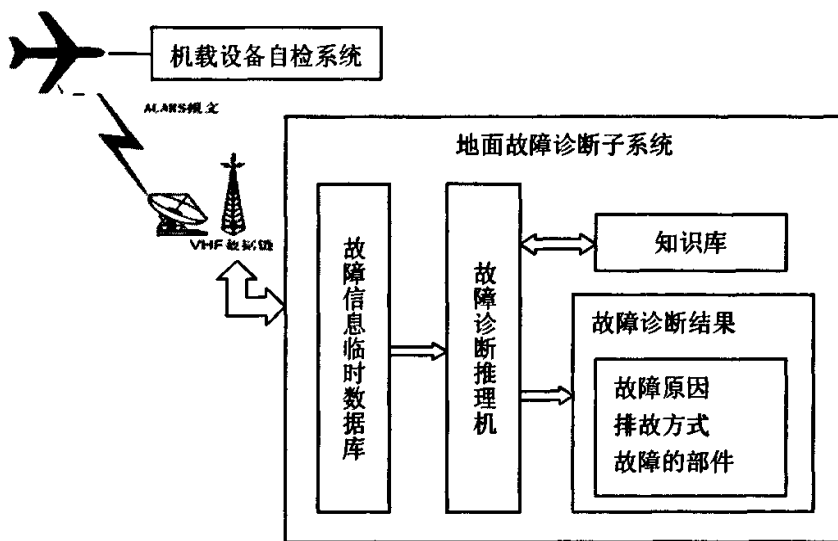


图 3-1 实时故障诊断系统体系结构结构

作为地面诊断子系统，它应该能够完成以下的工作：

- (1) 对飞行参数和 BIT 信息进行广泛深入的综合，采用智能的方法实现故障诊断，将故障精确的定位到 LRU 或 LRM，及时发现潜在的故障并为飞

机维修提供支持方案。

- (2) 能够快速、较为准确的从历史案例中找到与现存故障最相匹配的案例，来指导机上、外场、进站的维护与维修工作。
- (3) 针对飞行数据的复杂性、关联性，数据表现形式的各异性，历史飞行数据的巨量性，制定相应的管理策略，以适应系统的开发速度、运行速度和功能的要求。
- (4) 针对飞机故障的多样性，历史故障库的不断完善要有一个合理的学习机制来管理库中知识。
- (5) 对于诊断出的结果要有一个高效的修改与评价机制，对结果进行评估，以达到案例库的不断完善。

在地面子系统中, 需建立以下数据库:

- (1) 不断扩大的飞机数据库, 用于存储各架飞机的历史故障数据、历史飞行参数。维护子系统可以根据这些数据完成飞机健康趋势预测和部件寿命预测。
- (2) 不断扩充的故障诊断和预测专家知识库, 用于存储故障判定准则数据、专家诊断知识、故障推理规则等。

### 3.2 系统总体框架

按照系统设计的目标，设计系统框架如图 3-2 所示。

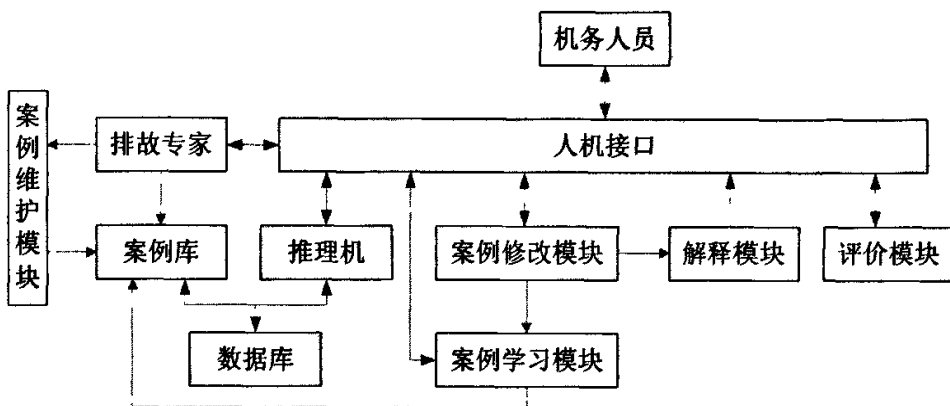


图 3-2 基于 CBR 故障诊断专家系统结构框架

框架中涉及到的各个模块介绍如下：

- (1) 推理机：采用最近邻算法结合距离测距的相似度算法进行案例匹配的推理，在相似度满足条件的前提下，保证匹配成功的历史案例尽量和目标案例的相同的征兆数最大。
- (2) 案例的修改模块：采用将诊断结果分为主和参考两类，比较各案例的不同，在主案例的基础上进行修改；采用基于 RBR 方法，对相同 BIT 信息下的过剩征兆采用 RBR 进行推理，而对于匹配征兆则尽量采用目标案例的征兆值，或做少量的更改。另外结合评价模块对于诊断结果进行再修改，达到与实际最一致。
- (3) 案例学习模块：确定案例学习的几种情况：
  - 1) 案例库中不存在待诊断故障的 BIT 信息
  - 2) 目标案例中的征兆数量与案例库中现有的征兆数量不同
  - 3) 目标案例中的征兆值不同于案例库中的征兆值
  - 4) 修改后的案例
  - 5) 专家评价后的案例
- (4) 解释模块：将诊断出的案例进行分发或是做历史参考。
- (5) 案例维护模块：其中一个功能是利用粗糙集理论对征兆的权值进行调整，另外约简掉一些与对应故障无关的征兆。定期维护，也可由用户随时进行维护。另一个功能是将案例转换为数据库能够存储的方式，把领域专家知识或是历史经验按照预定的数据格式存储到历史案例库中，供案例匹配使用。
- (6) 评价模块：系统对诊断匹配出来的历史案例进行评价，是否与实际排故操作一致，差别有多大，差别在哪里，做出具体的修改意见，作为专家系统对评价过的案例进行学习。
- (7) 人机接口：是提供给用户（机务人员）进行实时查询某架飞机目前运行情况的接口，它提供了操作接口，同时也给用户提供了信息查询的功能，令使用者可以直观的看到故障诊断的结果。同时人机接口的另一个功能就是通过与领域专家或是机务人员进行交互，让其对诊断进行评价和添加新知识。
- (8) 数据库：是专家系统推理工作过程中，在计算机内存中的工作区，



它供应用程序随时读取需要的数据或是将计算的中间结果写入工作区。

根据以上的介绍，并且在仔细调查飞机外场飞机故障诊断与排除过程的基础上，得到该系统的数据流程，如图 3-3 所示。

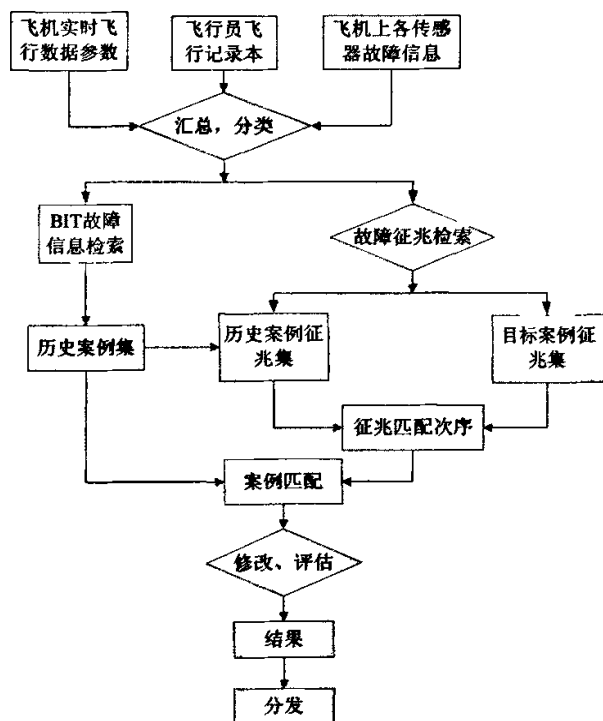


图 3-3 数据流程图

通过对基于 BIT 信息的故障诊断的内容和数据流程分析，设计的数据项和数据结构如下：

- (1) 登陆用户基本情况。其数据项有部门、部门编码等。
- (2) 历史案例的基本组成。其数据项有案例编号、飞机型号、故障发生时间、BIT 信息、发生故障的系统或部件、故障原因、排除方法、FIM 手册对应编号、AMM 手册对应编号等。
- (3) BIT 信息管理。其数据项有 BIT 信息、信息编号等。
- (4) 案例检索管理。其数据项有案例编号、案例被调用次数等。
- (5) 征兆检索管理。其数据项有 BIT 信息、征兆 1、征兆 2 等。

- (6) 征兆权值管理。其数据项有 BIT 信息, 征兆 1、征兆 2 等。
- (7) 征兆内容管理。其数据项有 BIT 信息, 征兆 1、征兆 2 等。
- (8) 征兆具体值管理。其数据项有 BIT 信息, 征兆 1、征兆 2 等。
- (9) 飞机信息管理。其数据项有飞机型号、入场时间、飞行总时间、上次飞行时间、上次故障时间、上次故障位置等。

通过分析系统中涉及到的数据流程、数据项和数据结构, 可以使设计思路更为清晰, 同时也为后续的进行数据库设计打下基础。

在系统的整体设计中案例的匹配算法以及征兆权值确定是该系统的设计难点和核心问题, 论文将在后续的章节中详细介绍。

### 3.3 案例的故障征兆的权值确定

权重系数的大小反映了在案例相似性评估中各特征属性的相对重要程度, 反映了专家对领域知识的理解, 是专家经验和决策者意志的体现。它在相当程度上决定了案例的检索精度, 取值的好坏将直接影响到评估结果的好坏。常用的赋权方法有: 专家咨询法、成对比较法、调查统计法、无差异折衷法以及相关分析法等等。在上述方法中, 前四种方法一般是在领域专家先验知识的基础上通过事先假设或采用平均权的办法来确定特征属性的权重值。最后一种方法基于一种统计的方法, 相对于前四种有一定的进步。针对技术应用案例的特点, 对不同框架我们采用不同的方法确定属性权重。

而应用到本系统中, 利用权值的大小来反映在特定的 BIT 信息(特定故障)下, 某个征兆对于这个 BIT 信息的作用的大小。由于飞机机载各设备交联关系复杂, 目前所积累下来的一些故障征兆不能完全表达它们所对应的故障, 按照粗糙集的理论, 对于机载设备的故障诊断我们可以采用粗糙集的相关理论来处理故障所对应的故障征兆。因此本系统利用粗糙集的理论来确定各故障征兆对应的权值。

#### 3.3.1 粗糙集理论

粗糙集理论<sup>[17]</sup>(Rough Set Theory, 简称 RS 理论)是一种新的处理模糊和不确定性知识的数学工具, 它是波兰科学家 Pawlak 和来自波兰科学院、华沙大学的

一些逻辑学家,在研究信息系统逻辑特性的基础上,于 20 世纪 70 年代提出来的。他们借鉴了逻辑学和哲学中对不精确和模糊的各种定义,针对知识库,提出了不精确范畴等概念,并在此基础上形成了完整的理论体系——粗糙集理论。其主要思想就是保持分离能力不便的前提下,通过知识约简,导出问题的决策和分类规则,在最初的几年中,由于大多数研究论文是用波兰文发表的,故未引起国际理论界和工程应用研究人员的重视,研究地域仅限于东欧各国。1991 年 Pawlak 的第一本关于粗糙集理论的专著“*Rough set: theoretical aspects of reasoning about data*”和 1992 年 Slowinski 主编的“*Intelligence decision support: handbook of application and advances of rough sets theory*”的出版,奠定了粗糙集理论的基础,有力的推动了国际粗糙集理论应用的深入研究。

作为一种研究不确定、不完整信息系统分类问题的数学工具,RS 理论有许多特点<sup>[21]</sup>,如:

- (1) RS 理论在数学上非常严密,有一整套处理数据分类问题的数学方法,特别当数据是不精确、不完整和不确定的时候。
- (2) RS 理论无需提供除问题所需处理的数据集合之外任何先验信息。(比如概率统计学中所需要的概率分布,模糊集理论中所必需的隶属函数等)。这是和模糊理论、证据理论的最主要区别。
- (3) RS 理论将知识定义为不可分辨关系的族,因此,知识有比较清晰的数学含义,可以很方便的用数学方法来进行分析和处理。
- (4) 可发现数据中隐含的模式和关系,对数据进行约简,评价数据的重要性,从数据中产生规则。
- (5) 粗糙集是一种软计算方法。软计算(soft computing)的概念是由模糊集创始人 Zadeh 提出的。传统的计算方法即所谓的硬计算。使用精确、固定和不变算法来表达和解决问题,而软计算的指导原则是利用所允许的不精确、不确定性和部分真实性得到易于处理、鲁棒性强和成本较低的解决方案,以便更好的与现实系统相协调。软计算的主要工具包括粗糙集、模糊逻辑、神经网络、概率推理、遗传算法与其他进化优化算法、混沌理论等。

目前 RS 理论已经引起越来越多的科研人员的关注。

### 3.3.1.1 粗糙集基础概念

设  $U$  是感兴趣的对象组成的非空有限集合, 称  $U$  为论域 (universe), 对于任何子集  $X \subset U$ , 称为  $U$  中的一个概念。为规范化起见, 认为  $\emptyset$  也是一个概念, 并且  $U$  中的任何概念簇, 称为  $U$  的抽象知识, 简称为知识。

对于子集  $X, Y \subseteq U$ ,  $X$  和  $Y$  由属性  $R$  是不可分辨的, 表示子集  $X$  和子集  $Y$  都属于  $R$  的一个范畴, 称为基于  $R$  的不可分辨关系,  $R$  被称为不可分辨关系 (indiscernibility relation), 它是在论域  $U$  上的一种等价关系, 用  $[X]_R$  表示。  $x$  和  $y$  在  $U$  上是不可分辨的, 是指有属性  $P \subseteq R$ , 对  $x, y \in U$ , 对任意的属性  $a \in P$ , 有  $f(x, a) = f(y, a)$ , 称为  $P$  上的不可分辨关系 (等价关系), 记为  $ind(P)$ 。

给定知识库  $K = (U, R)$ , 对于每个子集  $X \subseteq U$  和一个等价关系  $R \in ind(K)$ , 根据  $R$  的基本集合的描述来划分集合  $X$ , 定义下面两个子集:

$$R_-(X) = \bigcup \{Y_i \in U \mid Ind(R) : Y_i \subseteq X\} \quad 3-1$$

$$R^-(X) = \bigcup \{Y_i \in U \mid Ind(R) : Y_i \cap X \neq \emptyset\} \quad 3-2$$

$R_-(X)$  为  $X$  的下近似空间,  $R^-(X)$  为  $X$  的上近似空间。下近似为所有那些被包含在  $X$  里面的等价类的并集; 上近似为所有那些与  $X$  有交的等价类的并集。

定义了上、下近似空间, 就可利用此概念定义边界域

$$Bn_R(X) = R^-(X) - R_-(X) \quad 3-3$$

有

(1) 当边界域为空, 即  $R_-(X) = R^-(X)$ ,  $X$  为  $R$  的可定义集

(2) 当边界域不为空, 即  $R_-(X) \neq R^-(X)$ ,  $X$  为  $R$  的粗糙集

正域:  $Pos_R(X) = R_-(X)$ , 负域:  $Neg_R(X) = U - R_-(X)$

### 3.3.1.2 数据离散化

运用 Rough Set 理论处理决策表时, 要求决策表中的值用离散 (如整型、字符串型、枚举型) 数据表达。如果某些条件属性或决策属性的值域为连续值 (如浮

点型数表达), 则在处理前必须进行离散化处理, 而且, 即使对于离散数据, 有时也需要通过将离散值进行合并(抽象)得到更高抽象层次的离散值, 这是 Rough Set 理论中的一类重要研究课题, 它是粗糙集理论分析的其他环节(属性约简、值约简、重要性计算)的前提条件。

离散化本质上可以归结为利用选取的断点来对条件属性构成的空间进行划分的问题, 把这个  $n$  ( $n$  为条件属性的个数) 维空间划分成有限个区域, 使得每个区域中的对象的决策值相同。选取断点的过程也是合并属性值的过程, 通过合并属性值, 减少属性值的个数, 减小问题的复杂度, 这也有利于提高知识获取过程中所得到的规则知识的适应度。

决策表  $S = \langle U, R, V, f \rangle$ ,  $R = C \cup \{d\}$  是属性集合, 子集  $C$  和  $\{d\}$  分别称为条件属性集和决策属性集,  $U = \{x_1, \dots, x_n\}$  是有限的对象集合即论域。

下面列出了一些目前常用的离散化算法:

- (1) 等距离划分算法。此种离散化算法是在每个属性上, 根据用户给定的参数来把属性值简单地划分为距离相等的断点值, 不考虑每个断点段中属性值个数的多少。假设某个属性的最大属性值为  $x_{\max}$ , 最小属性值为  $x_{\min}$ , 用户给定的参数为  $k$ , 则断点间隔为  $\delta = (x_{\max} - x_{\min}) / k$ 。以此得到该属性上的断点为  $x_{\min} + i\delta, i = 0, \dots, k$ 。这些断点之间的距离相等。
- (2) 等频率划分算法。此种离散化算法是根据用户给定的参数  $k$  把  $m$  个对象分成段, 每段中有  $m/k$  个对象。假设某个属性的最大属性值为  $x_{\max}$ , 最小属性值为  $x_{\min}$ , 用户给定的参数为  $k$ , 则需要将这个属性在所有实例上的取值从小到大排列, 然后进行平均划分为  $k$  段即得到断点集。每两个相邻断点之间所包含的属性值的个数是相等的。
- (3) Naive Scaler 算法。Naive Scaler 算法如下:  
对每一个属性  $a \in C$ , 进行下面的过程:
  - 1) 根据  $a(x)$  的值, 由小到大排列实例  $x \in U$ ;
  - 2) 从上到下扫描, 设  $x_i$  和  $x_j$  代表两个相邻的实例:

如果  $a(x_i) = a(x_j)$ ，则继续扫描；

如果  $d(x_i) = d(x_j)$ ，即决策相同，则继续扫描；

否则，得到一个断点  $c$ ， $c = (a(x_i) + a(x_j)) / 2$ 。

- (4) 基于属性重要性的离散化算法。属性的重要性是建立在属性的分类能力上的，为了衡量条件属性的重要性程度，我们可从表中删除这个属性，再来考察信息系统的分类会产生怎样的变化：如果去掉某属性会相应地改变分类，则说明该属性重要（改变的程度越大，重要性越高）；反之说明该属性地重要性低。算法步骤如下：

1) 首先根据条件属性的重要性由小到大对条件属性  $V_i (i=1, \dots, n)$  进行排序；在属性重要性相同的情况下，按条件属性断点个数由多到少对条件属性进行排序。

2) 对每个属性  $V_i \in A$  进行下面的过程：

3) 对属性  $V_i$  中的每一个断点  $C_j (j=1, \dots, i_j)$ ，考虑它的存在性：把信息系统中与  $C_j$  相邻的两个属性值的较小值改为较大值，如果信息系统不引入冲突，则  $C_{V_i} = C_{V_i} \setminus \{c_j\}$ ；否则，把修改过的属性值还原。

这个算法通过对每一个断点进行判定，去掉冗余的断点，从而得到简化的信息系统。

### 3.3.1.3 属性的约简

实践中，经常要在保持知识库中初等范畴的情况下消去冗余基本范畴，进行知识的约简。

- (1) 令  $R$  为一等价关系簇，并且  $r \in R$ ，如果

$$Ind(R) = Ind(R - \{r\}) \quad 3-4$$

称  $r$  为  $R$  中可省略的，否则  $r$  为  $R$  中不可省略的。通俗的说， $R$  是表达我们研究的对象的属性集合，在近似表达中有一些特征作用不大，可以将这些属性删除而不影响我们对对象的表达，去掉冗余属性  $r$  后，剩下的

属性集仍然保留其等价关系。

(2) 对于属性子集  $P \in R$ , 若存在  $Q = P - r, Q \subseteq P$ , 使得  $Ind(Q) = Ind(P)$ , 且

$Q$  为最小子集, 则  $Q$  为  $P$  的约简, 否则为  $Red(P)$ 。一个属性集合可以有多种化简。

(3)  $P$  中所有约简属性集中都包含的不可省略关系的集合, 即约简集  $Red(P)$

的交集为  $P$  的核, 它是表达知识必不可少的重要属性集。表示为

$Core(P)$ , 即  $Core(P) = \cap Red(P)$ 。

实际上, 一般产生约简的方式是逐个向核中添加可省略的属性, 并进行检查。由于可省略的属性关系集合的幂集的基数是多少, 就有多少种添加的方式; 最好的情况是所有不可省的属性关系集合本身就是约简, 此时的约简是惟一的。所以, 计算所有约简与计算一个最佳约简 (比如定义为关系最少) 都是  $NP$  难题。

#### 3.3.1.4 知识的相对约简和相对核

在应用中, 一个分类相对于另一个分类的关系非常重要。

令  $P$  和  $S$  为  $U$  中的等价关系,  $S$  的  $P$  正域记做  $Pos_P(S)$ ,

$$Pos_P(S) = \bigcup P_-(S) \quad 3-5$$

对于  $U|P$  的分类,  $U|S$  的正域是论域中所有通过用分类  $U|P$  表达的知识能够确定划入  $U|S$  类的对象的集合。

当  $Pos_P(S) = Pos_{P-r}(S)$  时, 称  $r \in P$  为  $P$  中  $S$  可省略的, 否则,  $r$  为  $P$  中  $S$  不可省略的。当  $P$  中每一个  $r$  都为  $S$  不可省略的, 则称  $P$  为  $S$  独立的, 当  $P-r$  为  $P$  的  $S$  独立子族, 且  $Pos_P(S) = Pos_{P-r}(S)$ , 则族  $P-r$  为  $P$  的  $S$  相对约简。  $P$  中所有  $S$  不可省略原始关系族, 即约简集  $Red_S(P)$  的集合交, 称为  $P$  的  $S$  核, 记为  $Core_S(P)$ 。即  $Core_S(P) = \cap Red_S(P)$ , 其中  $Red_S(P)$  为  $P$  中所有  $S$  约简族。

### 3.3.1.5 知识的依赖性度量

从一个给定知识中导出另一知识,我们必须研究属性之间的依赖性关系。当知识  $Q$  中的所有初等范畴可以用  $P$  中的某些初等范畴定义时,则知识  $Q$  依赖于  $P$ 。

为了度量知识的依赖性,我们定义如下

$$k = r_p(Q) = \text{Card}(\text{Pos}_p(Q)) / \text{Card}(U) \quad 3-6$$

当  $k=1$  时,称  $Q$  是由  $P$  全可导的;论域的全部元素可通过知识  $P$  划入  $U|Q$  的初等范畴;

当  $k=0$  时,称  $Q$  是全不可导的;论域中没有元素可通过知识  $P$  划入  $U|Q$  的初等范畴;

当  $0 < k < 1$  时,称  $Q$  是粗可导的;论域的全部元素只有属于正域的元素可通过知识  $P$  划入  $U|Q$  的初等范畴。

在一个知识系统中,不同的属性具有的重要性程度,一般是由领域专家给出的权值来确定的,具有很强的主观色彩;但在粗糙集中只需通过客观的数据计算,就可以获得。方法是从知识中去掉一些属性,再考察分类度  $r$  的变化,如果分类产生变化,说明重要性高,反之则低。例如对于属性子集  $P^* \subseteq P$  的重要性,我们就可以利用依赖度

$$r_p(Q) - r_{p-p^*}(Q) \quad 3-7$$

来衡量。

### 3.3.2 利用粗糙集理论确定征兆权值

利用粗糙集的关于属性依赖度来确定某个属性的重要性。首先针对本系统构建一个通用的决策表,决策表中的内容是特定 BIT 信息下的一些历史案例。在这个决策表中,定义论域  $U$  为案例的序列号,条件属性  $C$  定义为每个案例对应故障征兆的取值,决策属性  $D$  定义为每个案例的排故方式(保留、更换、维修)。这样,我们可以得到一个关于某个特定 BIT 故障信息的决策表,通过计算其依赖度来确定每个故障征兆属性的重要性,从而就确定了每个属性的权值,为后续的案例匹



配打下基础。

现以 BIT 故障信息为 STANDBY ALTIMITER PROBLEM 为例,从历史案例库中取出 10 个历史案例。以飞行参数作为条件属性,以维修方法作为决策属性。其中飞行参数集为  $X=(x_1,x_2,x_3,x_4)$ ,维修方法集  $D=(e)$ ,  $U$  表示案例集。经离散化处理后,便形成了一张决策表,如表 3-1 所示。

表 3-1 决策表

$U$	$x_1$	$x_2$	$x_3$	$x_4$	$e$
1	1	2	0	1	1
2	1	1	0	1	1
3	2	0	0	1	0
4	0	0	1	2	1
5	2	1	0	2	1
6	0	0	1	2	2
7	2	0	0	1	0
8	0	1	2	2	1
9	2	1	1	2	2
10	2	0	0	0	0

表 3-1 中的各属性离散量表示的数值区间如表 3-2 所示:

表 3-2 数据离散化取值区间

属性	属性的量化值		
	0	1	2
$x_1$	OFF	ON	不确定
$x_2$	[0.437,0.496)	[0.496,0.555)	[0.555,0.614]
$x_3$	[427,459)	[459,491)	[491,523]
$x_4$	[9500,9790)	[9790,10080)	[10080,10370]

按照 3.3.1.5 节所介绍的计算依赖度的方法,可以得到计算各属性重要性的步骤如下:

- (1) 确定属性集  $R$  (条件属性);
- (2) 确定决策属性  $X$ ;
- (3) 利用公式 3-1 计算  $X$  的  $R$  下近似;
- (4) 利用公式 3-6 计算  $X$  对于  $R$  的依赖性;
- (5) 分别从  $R$  中减掉一个属性  $x_i$ , 计算  $X$  的  $R-x_i$  下近似;
- (6) 利用公式 3-6 分别计算  $X$  对于  $R-x_i$  的依赖性;
- (7) 利用公式 3-7 计算属性  $x_i$  的重要度。

由以上步骤可以得到该案例中各属性的重要性为:

$$imp_{x_1} = r_X - r_{X-x_1} = 0 \quad imp_{x_2} = r_X - r_{X-x_2} = 0$$

$$imp_{x_3} = r_X - r_{X-x_3} = 0.2 \quad imp_{x_4} = r_X - r_{X-x_4} = 0.3$$

可以从计算结果看到, 属性  $x_1, x_2$  是可以简约的, 但是考虑到此案例属性集个数的限制以及我们对该系统故障与飞行参数信息之间的对应关系不是很明了, 所以在此案例中我们未对案例的属性集做约简处理, 而是将可以简约的属性  $x_1, x_2$  的重要性赋予一个很小的值 0.0001, 这样既对最终的案例匹配影响很小(可忽略), 而且也不会减少各案例的信息量, 在客观上保持案例的一致性。

## 3.4 案例的检索推理

### 3.4.1 案例的表示与索引

案例的检索的好与坏不光与其检索的算法有关系, 而且与事例库中所存储的案例的表示索引有关, 本系统利用了基于框架表示法的案例表示方法。

框架是一种描述所论对象属性的数据结构。所论的对象可以是一个事物、一个事件或者一个概念。一个框架由若干个“槽”组成, 每个“槽”又可划分为若干个“侧面”。一个槽用于描述所论及对象的某一个方面, 一个侧面用于描述相应槽的一个

属性。槽和侧面所具有的属性值分别称为槽值和侧面值。槽值可以是逻辑型或数字型的，具体的值可以是程序、条件、默认值或是一个子框架。

一个框架可以由框架名、槽、侧面和值四部分组成。

针对机载设备 BIT 信息、可见的故障征兆和飞行状态参数所表示的故障模式，利用框架表示法将其表示成事例库中不同的案例。则一个飞机故障信息案例可表示为表 3-3 所示格式。

表 3-3 飞机故障信息案例表示

框架名：BIT 信息— $x$				
槽名	侧面 1/值	侧面 1/值	.....	侧面 $n$ /值
案例编号	$x$	NULL	NULL	NULL
飞机型号		NULL	NULL	NULL
发生时间		NULL	NULL	NULL
BIT 信息		NULL	NULL	NULL
故障 LRU	LRU1	LRU2	.....	LRU $n$
原因分析		NULL	NULL	NULL
排除措施		NULL	NULL	NULL
FIM_NO				
AMM_NO				

则一个历史飞行故障信息就可表示成如表 3-3 所示的框架，框架的个数与历史飞行故障信息的个数相同。框架名中分别对不同的 BIT 信息做不同的索引值， $x$ 表示在同一个 BIT 信息下的案例的编号。

3.4.2 建立检索推理树

首先要利用粗糙集中所述的属性重要性的理论，计算框架中各属性的重要性，删掉那些重要度为0的属性及其数值，并且按照属性的重要性降序排列。

CBR检索过程是通过检索事例库中的征兆集来和用户输入的案例（目标案例）的征兆集匹配，这就启发我们可以按照案例征兆集中的各个征兆的重要程度，组

成如同故障树一样的树结构,树的根节点为某系统故障时对应的BIT信息,因为BIT信息是本系统中故障诊断的启发性信息,通过确定BIT信息可以缩小诊断的范围,提高故障诊断的速度。通过计算各个可见征兆和飞行参数的重要度,确定每层节点的节点值。具体的步骤如下所示:

- (1) 利用3.3节介绍的计算重要性方法计算各征兆重要性;
- (2) 确定根节点BIT信息;
- (3) 将重要度最小的属性置于下一层的右节点,剩下的属性置于同一层的左节点;
- (4) 重复步骤(3)的过程,直到不能区分每个属性的重要度为止,树建立完毕。

流程图如图3-4所示。

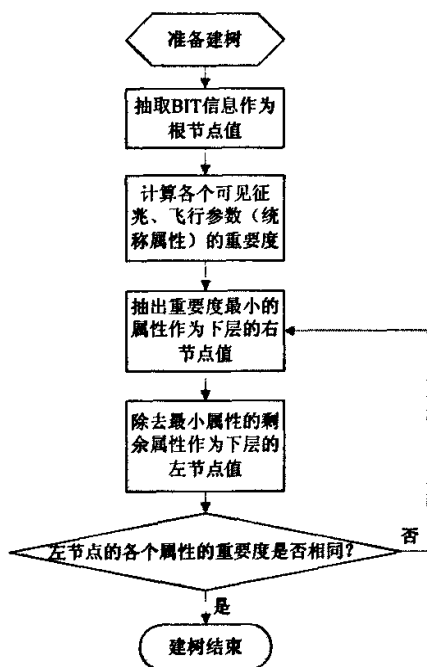


图3-4 建树流程图

建立的检索树是针对一种 BIT 信息而建立的,针对不同的 BIT 信息对应有不同的检索树,这些检索树统一存储在一个和 BIT 信息对应的表结构中,供检索时使用。随着历史案例库中的案例数量的增加,还应该定时对相同 BIT 信息下的检索树重新建立其检索结构。

现在可以以 3.3 节中的 BIT 信息为例,通过组织它的相关历史案例计算各故障征兆的重要度之后得到一个征兆检索树如图 3-5 所示。

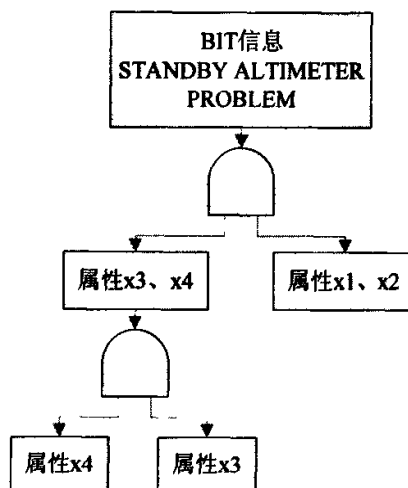


图 3-5 建立的征兆检索树

一旦建立了这个征兆检索树,在征兆检索过程中,系统就会首先检索这个检索树的最下层的左子树,然后检索右子树,继而再向上推一层检索,直到根节点为止。在同一个子节点中如果存在多个征兆,可以按照以下两个策略进行检索:

- (1) 按照其头字母的先后顺序进行检索;
- (2) 按照其属性的存储顺序进行检索;

### 3.5 案例的匹配

本系统的案例匹配采用两级相似匹配的策略,首先进行故障征兆的相似匹配,然后再进行整个案例的相似匹配,经过这样两级匹配出的案例诊断结果数量上会相应的减少,但在案例的正确性上有很高的保证。同时也可以预见的是,采用这样的匹配方式对于系统的资源利用是相当大的,系统会大量的学习案例,进行历史案例库的填充,但是也应该看到对于这样一个刚刚建立的系统而言,能够大量的填充案例对于系统的快速成熟应该是有好处的。这样不光可以使系统中源案例的粒度变小,同时可以更多的收集案例,对于大量的案例,系统可以自动的进行一些规则的抽取,对于系统今后的发展是至关重要的。

### 3.5.1 故障征兆的相似匹配

在征兆匹配中,采用了类似基于距离测度的相似评判方法,目前基于距离测度的相似评判方法为大部分基于 CBR 系统所使用。

距离测度方法包括下面两种:

1) 欧式距离

2) 海明距离

现介绍海明距离方法如下:

海明距离:  $SIM(X,Y)=1-DIST(X,Y)=1-\sum_i w_i dist(x_i,y_i)$

式中  $SIM(X,Y)$  表示案例  $X$  和  $Y$  的相似度,  $x_i$  和  $y_i$  分别为  $X$  和  $Y$  的属性。  $w_i$  代表第  $i$  个属性的重要度,  $i=1,\dots,n$ ,  $n$  是案例属性的个数, 标准的  $dist(x_i,y_i)$  通常表示为:  $dist(x_i,y_i)=|x_i-y_i|/|\max_i-\min_i|$ 。

对于数字属性值,  $\max_i$  和  $\min_i$  分别代表案例的第  $i$  个属性的最大值和最小值。对于符号属性值, 如果  $x_i=y_i, dist(x_i,y_i)=0$ ; 否则,  $dist(x_i,y_i)=1$ 。当  $dist(x_i,y_i)=0, SIM=1$ , 这意味着两个案例是同一的。当  $dist(x_i,y_i)=1, SIM=0$ , 则表明两个案例完全不同。

通过效仿海明距离的计算方法, 确定本系统中所使用的故障征兆的匹配算法如公式 3-8 所示:

$$SIM(X,Y)=1-DIST(X,Y)=1-dist(X,Y) \quad 3-8$$

其中  $SIM(X,Y)$  表示故障征兆  $X$  和  $Y$  的相似度, 改变了  $dist(X,Y)$  表示形式, 表示为目标案例相对于源案例的距离:  $dist(X,Y)=|X-Y|$ , 其中  $X$  为源案例的征兆值,  $Y$  为目标案例的征兆值, 应该提出的是,  $X$  和  $Y$  所对应的征兆内容是一致的。设定一个阈值来约束得到的  $dist(X,Y)$  值是否满足条件。设定的阈值实际上是设定了目标案例征兆值相对于源案例征兆值的浮动范围, 这个范围的设定可以利用在 3.3 节中计算出来的征兆的重要度来确定。具体的确定办法是, 将这个阈值确定为征兆重要度的倒数, 这样, 一方面体现了重要度在匹配中的作用, 一方面又将重

要度的尺度作用于个体征兆的匹配中,可以知道,当某个待匹配的征兆重要度很大时,则系统对这个征兆的约束就会很强,因为重要度越大,它的倒数会越小,反映到它的浮动范围上就会很小;反之,约束就会变弱,因为浮动范围会很大。可以通过公式 3-9 直观的看到这一点:

$$dist(X,Y)=|X-Y|\leq 1/imp_{X(Y)} \quad 3-9$$

式中的  $dist(X,Y)=|X-Y|$  为待匹配征兆之间的差异,  $imp_{X(Y)}$  为征兆  $X(Y)$  的重要度。只有满足公式 3-9 所约束的条件,这个相应的征兆才可算做匹配,进行下一个征兆的匹配过程,这里确定的下一个征兆的匹配时按照 3.4 节中所论述到的检索树的检索过程确定到底对哪个征兆进行匹配计算。如果不能满足约束条件,则这个征兆所在的案例的其他征兆将不再进行匹配计算,直接到下一个案例进行征兆的匹配。

采用这种策略即可以保证案例的匹配的速度,又可保证案例匹配的高度一致性。

### 3.5.2 故障案例的相似匹配

在 3.5.1 节的相似度匹配的基础上,可以对整个案例的相似度进行计算。如果说 3.5.1 称为局部相似度的匹配的话,那么此时的匹配就可称为全局相似度的匹配。根据属性权重和属性相似度,用最邻近算法 (NNH) 可以得到案例级的全局相似度。全局相似度计算公式如下:

$$OS^{NN}(x_j, q) = \frac{\sum_{f=1}^l (w_f * Sim(x_{jf}, q_f))}{\sum_{f=1}^l w_f}$$

其中,  $OS^{NN}(x_j, q)$  表示历史案例 (源案例)  $x_j$  与新案例 (目标案例)  $q$  的全局相似度,  $Sim(x_{jf}, q_f)$  为历史案例  $x_j$  与新案例 (目标案例)  $q$  在征兆  $f$  上的相似度; 需要计算征兆  $f$  的个数, 征兆的权值为  $w_f$ 。

NNH 结合了领域知识, 大多数的 CBR 系统都采用该算法。NNH 成立的假设是两个案例之间的属性集是一样的, 且属性间是相互独立的, 存在合适的匹配规则

和程序。最简单的最邻近算法是采用加权平均的方法，将所有的属性的相似度过加权加总后就可以得到两个案例的相似度。

### 3.6 案例的修改

这里所论及的修改包括两个方面的内容：

(1) 对候选案例进行的修改过程

(2) 对多个案例进行重组修改过程

这两个方面分别对应不同的内容。

#### 3.6.1 候选案例的修改

这里所用的修改策略是采用被动式的修改方式，即采用专家评估给与修改意见，从而驱动系统去学习修改过的案例，为今后同等故障提供参考。

由于案例库中的案例规模还较小，对于从现有案例中提取规则的方法还没有基础。案例数量增大，案例中征兆的确定已被证明对于故障是有作用的，此时，可以通过 3.3 节中介绍的关于属性的约简的方法提取关于产生故障的规则，从而主动式的去驱动系统完成案例的修改，这些规则可以组成一个用于修改的数据集。当目标案例中包含的故障征兆在源案例故障征兆中无法检索到时，可以启动这个数据集对候选案例进行修改，从而给出满足目标查询的诊断结果。这里提到的数据集实际上是一个规则库，可以定期对案例库进行维护，从中提取规则来填充这个规则库。

#### 3.6.2 多案例的重组修改

这里提到的多案例重组可解决两个方面的问题：

(1) 对于系统可能接收到不止一个 BIT 故障信息，系统如何来控制这些信息进行故障诊断，最终对多个案例重组得到故障诊断结果。

(2) 对于系统接收到的一个 BIT 故障信息，系统诊断出的案例如果比较多，那么如何来对这些案例进行重组得到最终的故障诊断结果。

通过比较诊断出的案例的 FIM\_NO、AMM\_NO 字段内容，来反映各个案例之



间的不同,从而进行案例的诊断。因为不同的 FIM\_NO、AMM\_NO 会对应不同的故障隔离方式和不同的 LRU,这样便能体现出各个案例的诊断结果的不同,从而协助系统完成案例的修改。

### 3.6.2.1 多 BIT 信息下的案例组合

产生多 BIT 故障信息,可能确实存在每种 BIT 信息所描述的故障,而大部分是由于一个 BIT 故障信息(即真实故障)所引发的其他故障信息,这些虚假的故障信息应该对系统是有干扰的。但是,对于基于 CBR 的诊断方式,系统只是记录历史上发生过的案例,这些案例中就包含了多 BIT 信息下的诊断实例,所以我们只需要对每种 BIT 信息分别诊断,这样通过每种 BIT 信息诊断出的案例对于最终的诊断结果应该说都是有参考价值的,所以需要将这些案例进行重新组合,使其符合实际需要。这里采用的策略是按照将案例分主次来进行的,主次之分的确定,是按照每个匹配出来的案例与目标案例征兆值相同最多或是征兆项吻合最多的最主要,以最主要的结果作为修改的主案例,其他的案例作为修改的参考案例,对主案例进行修改。

### 3.6.2.2 单 BIT 信息下的案例修改

这里采用的策略与 3.6.2.1 节所用的方法类似,区别在于,这里的案例是对应的同一种 BIT 信息,仍然要按照主次来区分匹配出来的案例,确定源案例与目标案例征兆值相同数目最多的案例或是故障征兆相同数目最多的案例作为主案例,在这个案例的基础上进行修改。更改的内容主要集中在排故和故障原因两方面尽量提供给用户更完善的结果。

## 3.7 推理算法及有效性验证

### 3.7.1 故障诊断的推理算法

通过前面几节的深入研究与详细分析,系统的推理算法流程图可以如图 3-6 所示。

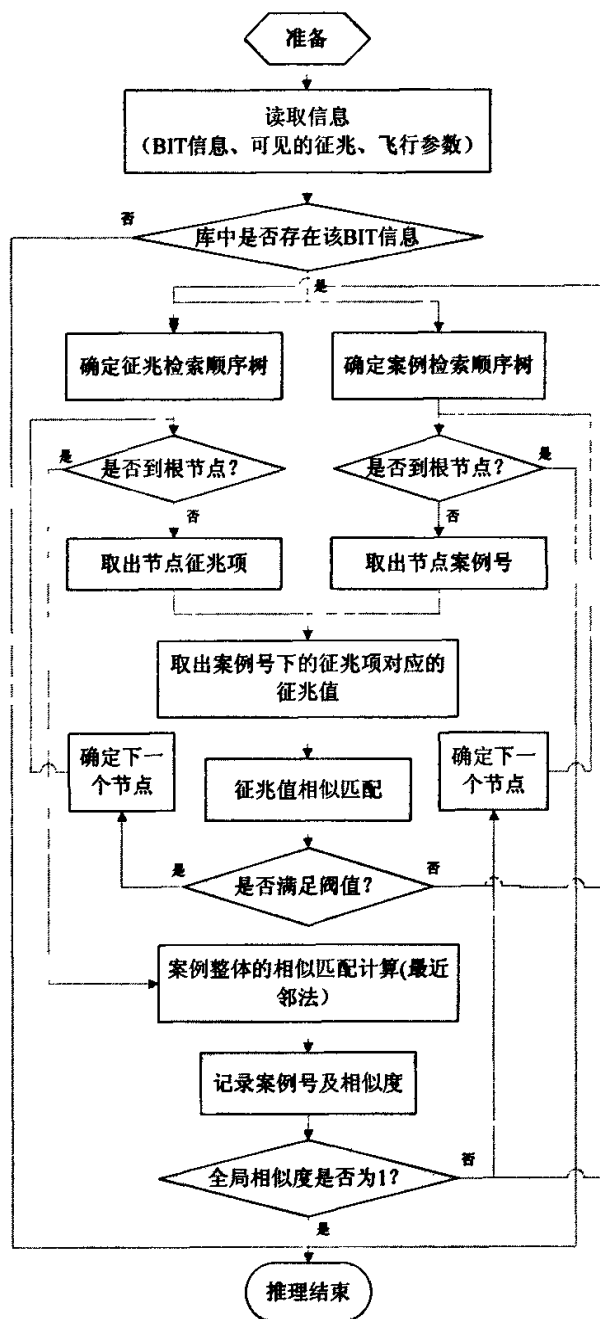


图 3-6 推理算法流程图

从图中可以看到，推理是以 BIT 信息为主要线索，首先通过 BIT 信息将征兆的检索树和历史案例的检索树组织起来，然后每次从征兆检索树中取出征兆项，同时从案例检索树中取出案例，以征兆项为索引值，取出案例对应的征兆值。首先将征兆值进行局部的相似匹配，即检验其与目标案例中的该征兆项对应的征兆值之间的距离是否满足阈值，如果满足进行下一个征兆项的匹配，如果不满足，

取出下一个案例,进行如同上一个案例一样的相似匹配过程。如果某个案例的局部相似性都满足条件,则进行这个案例的全局相似匹配,将计算的出的全局相似度与该案例同时记录下来。如果计算出的全局相似度为 1,则整个推理过程结束,表示通过推理找到了与目标案例完全一致的历史案例;如果全局相似度不为 1,则继续进行如上的案例局部以及全局的匹配,直到遍历整个案例检索树,推理过程结束。

### 3.7.2 推理算法有效性验证

按照流程图所描述的推理过程,下面将给出基于上述算法的有效性验证分析。这里只是针对相似匹配过程的验证,而对于修改,评价等由于不同的应用对象有不同的修改策略,而有可能在某种应用场合某个修改或是评价策略很合适,而将其照搬到另一个应用场合却根本无任何作用,所以这里只是针对相似匹配作验证。由于在本文介绍的推理策略中,增加了一级局部匹配过滤功能,验证的过程就是要检验这个功能的加入与否对于诊断结果的影响有多大,另外对于案例在匹配过程中征兆匹配的次数有多大影响。诊断过程的时间消耗问题,对于地面系统而言,在时间的要求上不像机载的诊断系统那样苛刻,所以相对机载,地面诊断系统的时间相对而言是很宽裕的。

在算法验证中,采用了历史案例库中的故障信息为 RADIO ALT-1 这样的 BIT 信息。这条信息代表内容是 FCC-A 或者 FCC-B 没有接收到一号无线电高度表的信号或是接收到的信号无效。这个 BIT 信息的故障征兆包括 DFCS BITE、LRRRA BITE、channel A、channel B、radio transmeter-1、pinB4-B2、pinA4-B3 这七项,在历史案例库中存在的和该 BIT 信息相关的所有案例共有 10 条,案例编号从 09-01~09-10,各征兆的重要度如表 3-4 所示。

表 3-4 征兆重要性表

征兆	DFCS BITE	LRRRA BITE	channel A	channel B	radio transmeter-1	pinB4-B2	pinA4-B3
重 要 性	0.6	0.6	0.7	0.7	0.9	0.8	0.8

各历史案例的每个征兆值对应如下表（表 3-5）所示。

表 3-5 案例征兆值表

案 例 \ 征兆	DFCS BITE	LRRA BITE	channel A	channel B	radio transmeter-1	pinB4-B2	pinA4-B3
09-01	1	0	1	0	0	0	0
09-02	1	0	0	1	0	0	0
09-03	1	0	0	0	1	1	1
09-04	1	1	0	0	0	0	0
09-05	1	0	0	0	1	1	0
09-06	1	0	0	0	1	0	1
09-07	1	1	0	0	1	1	1
09-08	1	0	1	0	1	1	0
09-09	1	0	1	1	0	0	0
09-10	1	0	0	1	1	0	1

备注：表中 0 代表无故障，1 代表故障存在

表中所示的案例值是经过将定性数据转变为定量数值之后的数据。

10 个历史案例的检索顺序如表 3-6 所示。

表 3-6 案例历史成功匹配次数表

案例	09-01	09-02	09-03	09-04	09-05	09-06	09-07	09-08	09-09	09-10
次数	3	1	0	0	1	4	1	0	3	0

目标案例 1（待诊断案例）如表 3-7 所示。

表 3-7 目标案例 1 征兆值表

征兆	DFCS BITE	LRRA BITE	channel A	channel B	radio transmeter-1	pinB4-B2	pinA4-B3
目 标 案例	1	0	0	0	1	1	0

现在对以下两种推理策略作实际对比：

策略 1：仅仅考虑案例的全局相似度策略。

策略 2：同时考虑局部相似度和全局相似度策略。

比较 1：目标案例存在于历史案例中。

通过试验比较征兆匹配的次数，案例的匹配个数，以及匹配结果，得到如下表（表 3-8）所示。

表 3-8 结果比较一览表

	征 兆 匹 配 次 数 (次)	案 例 匹 配 个 数 (个)	匹配结果（案例编号）
策略 1	70	10	09-05 、 09-07 、 09-08
策略 2	12	5	09-05

利用策略 1 得到的诊断结果是选取了相似度最大的三个案例。可以比较以下两种策略下的诊断结果，策略 1 要比策略 2 得到的结果多，但是在准确性方面确说明了策略 2 更准确一些。它能直接定位到最准确的案例。而且在前两个指标下，也能看的出来征兆的匹配次数上，策略 2 相对策略 1 有很明显的优势。可以设想，当历史案例很多的情况下，策略 2 能够显著的减少征兆的匹配的次数。

比较 2：目标案例不存在于历史案例库中。

目标案例 2（待诊断案例）如表 3-9 所示。

表 3-9 目标案例 2 征兆值表

征兆	DFCS BITE	LRRA BITE	channel A	channel B	radio transmeter-1	pinB4-B2	pinA4-B3
目 标 案例	1	1	0	0	1	1	0

如表 3-10 得到案例的比较结果。

表 3-10 结果比较一览表

	征 兆 匹 配 次 数 (次)	案 例 匹 配 个 数 (个)	匹配结果（案例编号）
策略 1	70	10	09-05 、 09-07 、 09-08
策略 2	25	10	无

可以看到由于上述的案例的征兆全都是定性的数据，表中给出的只是将定性的数据量化后的结果。在这种情况下，采用策略 2 是只有完全匹配与完全不匹配两种情况，不会出现如同策略 1 所给出的结果，这说明策略 2 的匹配精确性，同时也说明了该算法在处理定性（符号）数据的不足，这种不足体现在诊断结果上。

但是对于有定量数据的案例就不会出现类似的问题，因为对于定量数据它本身有浮动范围，这样它便不会像定性数据那样会产生分明的不同或是相同的界限。可以从下面的比较实例中得到验证。

比较 3：故障征兆中有数值型数据

在历史案例中关于故障信息 STANDBY ALTIMETER PROBLEM 的征兆包含有继电开关、马赫数、飞行高度、CDS 与备用指示器所显示的高度的差值这四项故障征兆。这四个征兆中，只有继电开关是一个定性的数据，它有关闭、连通、不确定三个状态，分别用 0、1、2 来数值化代表这三种状态。各个征兆的重要度如表 3-11 所示。

表 3-11 故障征兆权值表

征兆	继电开关	马赫数	飞行高度	高度的差值
重要度	0.0001	0.0001	0.2	0.3

历史案例库中与该故障信息相对应的故障案例有以下 12 个，如表 3-12 所示。

表 3-12 案例征兆值表

	继电开关	马赫数	飞行高度	高度的差值
10-01	1	0.574	10012	432
10-02	1	0.500	10037	427
10-03	2	0.465	9890	453
10-04	0	0.483	10109	471
10-05	2	0.527	10201	431
10-06	0	0.441	10370	490
10-07	2	0.459	10024	448
10-08	0	0.545	10275	510
10-09	1	0.464	10033	433
10-10	0	0.523	10259	529
10-11	1	0.498	10027	430
10-12	1	0.529	10370	473

目标案例的征兆值如表 3-13 所示。

表 3-13 目标案例征兆值

征兆	继电开关	马赫数	飞行高度	高度的差值
征兆值	1	0.473	10032	430

如表 3-14 得到案例的比较结果。

表 3-14 结果比较一览表

	征兆匹配次数 (次)	案例匹配个数 (个)	匹配结果(案例编号)
策略 1	48	12	10-10、10-11、10-12
策略 2	22	12	10-02、10-09、10-11

匹配结果中，策略 1 的结果是取相似度最大的前三个案例作为结果。而策略 2

的结果是匹配出来的局部相似度满足约束条件的案例，而未对全局相似度进行取舍。可见没有局部的约束的情况下，虽然策略 1 中的近邻算法中乘以了征兆权值，但是全局的相似度匹配仍在某种程度上不可避免的受到目标案例与源案例之间过大的征兆值差的影响，最终就会反映在诊断的结果上。而采用策略 2 却注意到了这种受过大的征兆值差带来的在结果上的影响，从而从局部就作约束，这样不能说完全避免了策略 1 遇到的这种问题，但是可以看得出是很有效的对这种效应进行了抑制。

通过上面的三个比较案例，可以清楚的看到策略 1 和策略 2 存在的不同是比较明显的，相对于策略 1，策略 2 不仅仅是能在诊断的结果上更加准确，同时当历史案例特别多的情况下，策略 2 还能够有效的减少征兆匹配的次数，案例匹配的次数，提高了整个系统的效率。

### 3.8 本章小节

本章主要对基于 CBR 的故障诊断专家系统的设计及其算法进行了讨论研究。首先分析了系统的设计目标，并据此设计系统的总体框架。继而在整体框架的基础上，分别就案例的表示、案例的相似匹配、案例属性的权值确定，案例的修改、案例的评价进行了深入的探讨，分别给出了具有可行性的解决方案。最后针对提出的故障诊断推理算法（3.7.1 节介绍算法），进行了推理验证，得到了令人满意的比较结果。



## 第四章 故障诊断专家系统的详细设计

故障诊断专家系统的设计包括以下两个方面的内容:

- (1) 案例数据库的设计
- (2) 专家系统功能软件的设计

数据库结构设计的好坏将直接对应用系统的效率以及实现的效果产生影响,好的数据库结构设计会减小数据的存储量,数据的完整性和一致性比较高,系统具有较快的响应速度,简化基于此数据库的应用程序的实现方法等<sup>[23]</sup>。在数据库开始设计的时候应该尽量考虑全面,尤其应该认真考虑用户的各种需求,避免浪费不必要的人力和物力。

专家系统的功能结构要合理,软件逻辑结构的好会能够影响软件执行的效率以及正确性。

### 4.1 案例数据库系统设计

#### 4.1.1 案例数据库概念结构设计

这一设计阶段是在需求分析(3.2节中已作详细分析)的基础上进行的,设计出能够满足用户需求的各种实体,以及它们之间的关系,为后面的逻辑结构设计打下基础。

根据3.2节的设计规划得出的实体有:部门实体、飞机实体、案例实体、BIT信息实体、故障征兆实体。各个实体具体的E-R图如图4-1~图4-5所示。

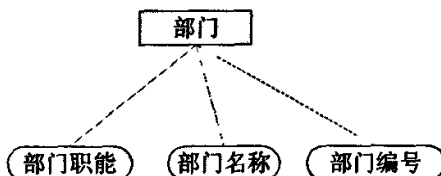


图 4-1 部门实体 E-R 图

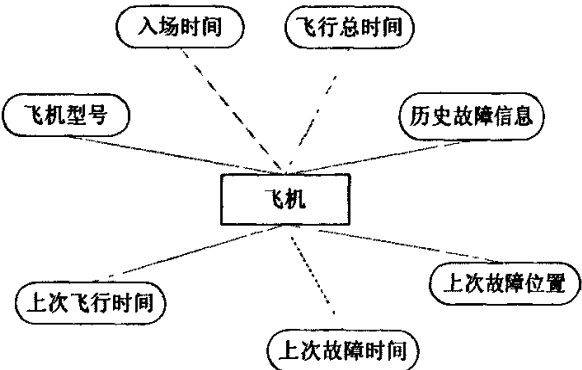


图 4-2 飞机实体 E-R 图

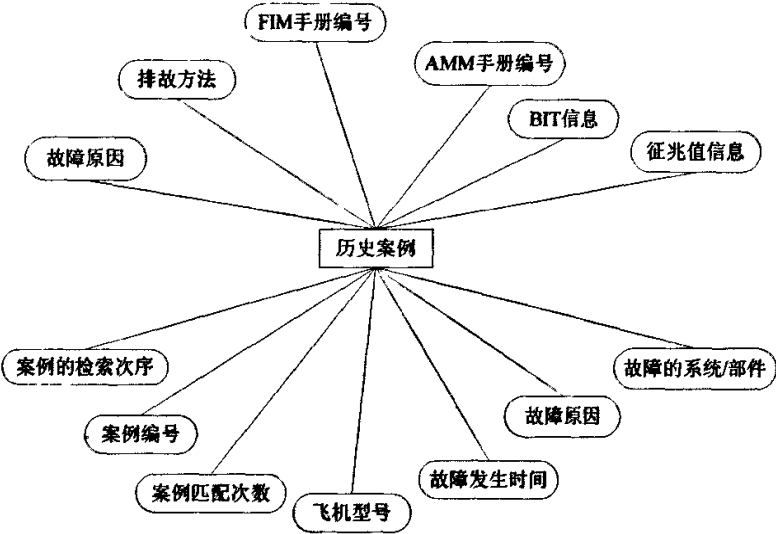


图 4-3 历史案例实体 E-R 图

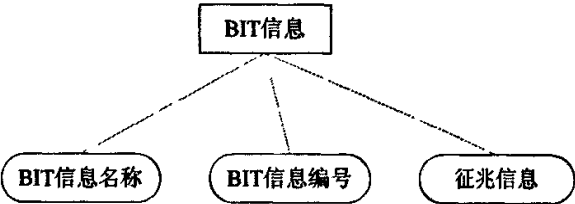


图 4-4 BIT 信息实体 E-R 图

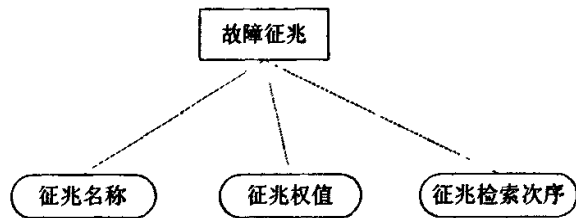


图 4-5 故障征兆实体 E-R 图

实体和实体之间的 E-R 图如图 4-6 所示。

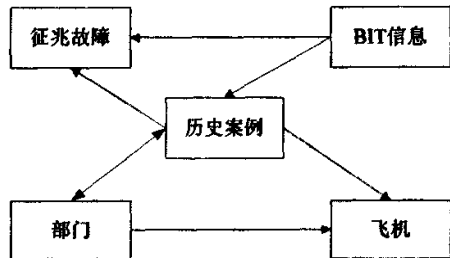


图 4-6 各实体之间 E-R 图

4.1.2 案例数据库逻辑结构设计

在上面的实体以及实体之间的关系的基礎上，形成数据库中的表格和各个表格之间的关系。

基于 BIT 信息故障诊断系统数据库中各个表格的设计结果如下面的几个表格所示。每个表格都表示数据库中的一个表。

表 4-1 历史案例表 History\_Case

列名	数据类型	可否为空	说明
Case_ID	VARCHAR(50)	NOT NULL	案例编号（主键）
Plane_Type	VARCHAR(10)	NOT NULL	故障飞机型号
AccurTime	DATETIME	NOT NULL	故障发生的时间
BIT_Info	VARCHAR(50)	NOT NULL	故障信息
Fault_Reason	TEXT	NOT NULL	故障发生原因
Troubleshooting	TEXT	NOT NULL	故障的处理方法

FIM_NO	VARCHAR(50)	NOT NULL	故障隔离手册对应排故章节
AMM_NO	VARCHAR(50)	NOT NULL	维护手册对应维护章节
Fault_LRU	VARCHAR(50)	NOT NULL	故障的现场可更换单元

表 4-2 BIT 信息索引表 BIT\_Index

列名	数据类型	可否为空	说明
BIT_Info	VARCHAR(50)	NOT NULL	BIT 信息（主键）
BIT_ID	CHAR(10)	NOT NULL	BIT 信息编号

表 4-3 历史案例检索的顺序表 Case\_Retrieve

列名	数据类型	可否为空	说明
Case_ID	VARCHAR(50)	NOT NULL	案例编号（外键）
Retrieve_time	INT(4)	NOT NULL	案例匹配成功次数

表 4-4 故障征兆（飞行参数）集表 Fault\_Symp

列名	数据类型	可否为空	说明
BIT_Info	VARCHAR(50)	NOT NULL	BIT 信息（外键）
Symp_1	VARCHAR(50)	NULL	故障征兆 1
Symp_2	VARCHAR(50)	NULL	故障征兆 2
Symp_3	VARCHAR(50)	NULL	故障征兆 3
Symp_4	VARCHAR(50)	NULL	故障征兆 4
Symp_5	VARCHAR(50)	NULL	故障征兆 5
Symp_6	VARCHAR(50)	NULL	故障征兆 6
Symp_7	VARCHAR(50)	NULL	故障征兆 7
Symp_8	VARCHAR(50)	NULL	故障征兆 8

Symp_9	VARCHAR(50)	NULL	故障征兆 9
Symp_10	VARCHAR(50)	NULL	故障征兆 10

表 4-5 故障征兆权值表 Fault\_Symp\_Imp

列名	数据类型	可否为空	说明
BIT_Info	VARCHAR(50)	NOT NULL	BIT 信息（外键）
Symp_1	VARCHAR(50)	NULL	故障征兆 1
Symp_2	VARCHAR(50)	NULL	故障征兆 2
Symp_3	VARCHAR(50)	NULL	故障征兆 3
Symp_4	VARCHAR(50)	NULL	故障征兆 4
Symp_5	VARCHAR(50)	NULL	故障征兆 5
Symp_6	VARCHAR(50)	NULL	故障征兆 6
Symp_7	VARCHAR(50)	NULL	故障征兆 7
Symp_8	VARCHAR(50)	NULL	故障征兆 8
Symp_9	VARCHAR(50)	NULL	故障征兆 9
Symp_10	VARCHAR(50)	NULL	故障征兆 10

表 4-6 故障征兆值表 Fault\_Symp\_Val

列名	数据类型	可否为空	说明
Case_ID	VARCHAR(50)	NOT NULL	案例编号（外键）
Symp_1	VARCHAR(50)	NULL	故障征兆 1
Symp_2	VARCHAR(50)	NULL	故障征兆 2
Symp_3	VARCHAR(50)	NULL	故障征兆 3
Symp_4	VARCHAR(50)	NULL	故障征兆 4
Symp_5	VARCHAR(50)	NULL	故障征兆 5
Symp_6	VARCHAR(50)	NULL	故障征兆 6
Symp_7	VARCHAR(50)	NULL	故障征兆 7
Symp_8	VARCHAR(50)	NULL	故障征兆 8
Symp_9	VARCHAR(50)	NULL	故障征兆 9

Symp_10	VARCHAR(50)	NULL	故障征兆 10
---------	-------------	------	---------

表 4-7 故障征兆检索顺序表 Fault\_Symp\_Retrieve

列名	数据类型	可否为空	说明
BIT_Info	VARCHAR(50)	NOT NULL	BIT 信息（外键）
Symp_1	VARCHAR(50)	NULL	故障征兆 1
Symp_2	VARCHAR(50)	NULL	故障征兆 2
Symp_3	VARCHAR(50)	NULL	故障征兆 3
Symp_4	VARCHAR(50)	NULL	故障征兆 4
Symp_5	VARCHAR(50)	NULL	故障征兆 5
Symp_6	VARCHAR(50)	NULL	故障征兆 6
Symp_7	VARCHAR(50)	NULL	故障征兆 7
Symp_8	VARCHAR(50)	NULL	故障征兆 8
Symp_9	VARCHAR(50)	NULL	故障征兆 9
Symp_10	VARCHAR(50)	NULL	故障征兆 10

从上述的各个表结构中，可以看到表 4-4~4-7 的结构差不多是一样的，但从具体的表的使用来说，它们从内部的数据到表的功用都是不同的，所以它们是互不相同的。

在表 4-4 故障征兆（飞行参数）集表 Fault\_Symp 中的每个字段下的内容为不同 BIT 信息下的不同故障征兆名称；表 4-5 故障征兆权值表 Fault\_Symp\_Imp 中存放的是与表 4-4 中 BIT 信息相对应的各个征兆的权值（重要度）；在表 4-6 故障征兆值表 Fault\_Symp\_Val 中存放的是和案例编号对应的各个案例的故障征兆具体值，这些故障征兆的顺序和表 4-4 中 BIT 信息的征兆顺序相对应；4-7 故障征兆检索顺序表 Fault\_Symp\_Retrieve 中存放的是按照重要度排序后的故障征兆的顺序，它与征兆检索顺序有关。

4.1.3 案例数据库物理实现

经过前面的需求分析和概念结构设计以后，得到数据库的逻辑结构。现在可

以在 SQL Server 2000 数据库系统中实现该逻辑结构。这里利用 SQL Server 2000 数据库系统中的 SQL 查询分析器实现。

在本系统的设计过程中,为了满足计算机管理的需要,采取了下面的一些原则:

- (1) 程序代码标准化, 软件统一化, 确保软件的可维护性和实用性。
- (2) 界面尽量简单化, 做到实用、方便, 尽量满足不同层次员工的需要。
- (3) 建立操作日志, 系统自动记录所进行的各种操作和操作结果。

## 4.2 专家系统的功能分析

### 4.2.1 专家系统功能分析

根据该专家系统设计目标, 系统中需要完成的功能主要有以下几点:

- (1) 支持不同用户群的登陆
- (2) 目标案例故障信息的读取
- (3) 历史案例库故障值表中数据的维护
- (4) 历史案例征兆的权值的确定
- (5) 历史案例征兆的检索树建立
- (6) 目标案例的相似匹配 (诊断过程)
- (7) 专家对诊断结果的评价
- (8) 诊断结果的修改模块
- (9) 新案例的学习
- (10) 故障诊断结果的显示和存储
- (11) 操作日志的管理

### 4.2.2 专家系统各功能模块关系

在系统分析的基础上, 结合 Visual C++ 6.0 程序编制的特点, 得到如图 4-7 所示的系统功能模块关系图。

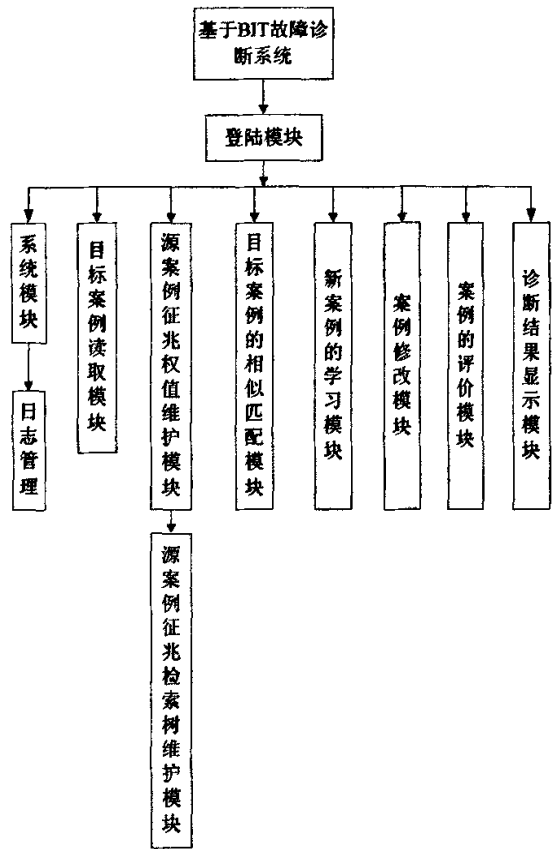


图 4-7 系统功能模块关系图

从图中可以看到，用户需要登录才能使用各种功能。目标案例的相似匹配模块是整个关系图中的核心模块，与它同一层次的其他模块都是作为辅助使匹配更准确。在相似匹配模块的基础上，由用户选择是否对案例进行评价和权值的维护，由程序决定是否进行学习、进行权值的维护或是案例的修改。最后系统将诊断结果自动存储并显示给用户。系统整个的功能运行都将记录在日志中以供系统的不断完善。

各模块的作用及功能实现将在下一节作具体介绍。

4.3 故障诊断专家系统功能模块设计

基于 BIT 信息的专家系统应用程序是利用微软的 Visual C++ 6.0、SQL server 2000 开发完成的。下面对程序中的几个功能模块的设计过程进行详细的介绍。



### 4.3.1 目标案例故障信息的读取程序模块

这个模块主要完成的功能就是从数据接收设备中读取从飞机上实时下载到地面的故障信息和飞行状态参数、故障征兆，以及它们各项的数值。并将这些数据存储在临时的数据库中。读取的流程如图 4-8 所示。

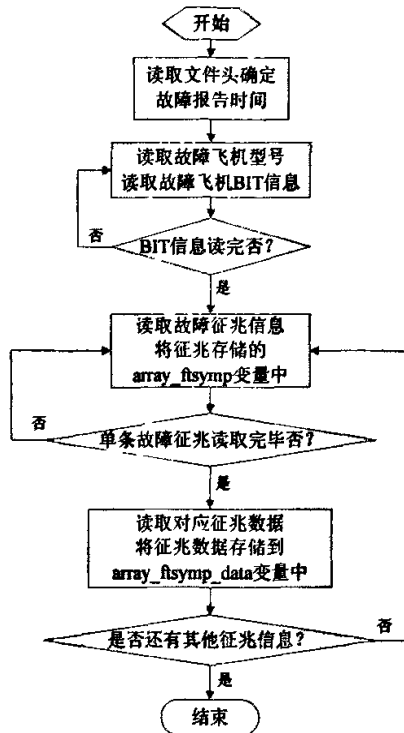


图 4-8 故障信息读取流程

程序中 `array_ftsymp`、`array_ftsymp_data` 是一个 `CStringArray` 类型的变量，这个变量可以动态的增加或删除其中的元素，完成了动态数组的功能。这就有利于读取的故障征兆，因为每次下载过来的故障征兆的数量是不同的，采用这种数据结构，对于处理这个问题是非常适合的。

这个变量是一个全局的变量，它的值将拷贝给一个临时变量用于后面的匹配操作，在匹配结束时要判断变量内部是否还有未被匹配的征兆，如果有，则要进入案例的学习程序。

### 4.3.2 目标案例的故障诊断（案例匹配）程序模块

这个程序模块的主要作用是根据目标案例的故障征兆，从历史案例库中检索和目标案例最相匹配的历史案例，进而给出用户故障诊断的结果。

其中采用征兆匹配过滤的方法，如果局部相似度不能满足约束条件，那么舍弃当前正在匹配的案例，转而匹配下一个历史案例；反之，如果满足约束条件，那么继续匹配同一个案例下的下一个征兆，如果所有的征兆全都满足约束条件的话，可以利用局部相似度计算案例的全局相似度，同时记录案例中与目标案例完全一致的征兆的个数。将匹配的案例存储到临时的数据库表 Case\_Similar 表中。这样，案例一经检索完毕，那么满足匹配条件的案例就会全部检索出来并且都存储到 Case\_Similar 表中。

这个程序的运行会涉及到数据库中历史案例表 History\_Case、BIT 信息索引表 BIT\_Index、历史案例检索的顺序表 Case\_Retrieve、故障征兆（飞行参数）集表 Fault\_Symp、故障征兆权值表 Fault\_Symp\_Imp、故障征兆值表 Fault\_Symp\_Val、故障征兆检索顺序表 Fault\_Symp\_Retrieve 和一张临时表案例相似度表 Case\_Similar 的读取、修改操作。

检索过程分为案例的检索过程与征兆的检索过程。征兆检索过程是按照前述的征兆检索树的检索顺序进行，征兆检索树的信息已经存储到了 Fault\_Symp\_Retrieve 表中，只需要按顺序读取表中数据，以读取的数据为参考，实现目标案例与源案例的匹配。案例的检索过程是通过读取 Case\_Retrieve 表中的数据来得到的，读取出来的数据是按照历史案例在当前故障诊断时刻之前曾经被相似匹配成功次数大小读取出来的，换句话说，就是第一个案例是匹配成功次数最多的案例，最后一个案例是匹配成功次数最少的案例。整个的检索过程是符合人的思考过程和人在排故过程中的操作方式的。

结果的显示是采用了两种方式进行，一个是在显示屏上实时将诊断结果提供给用户，另一个方式是将结果以文本的方式存储起来，供分发或是日后作为参考，以年月日架次作为文件的名称。

另外在显示的时候，根据不同登陆的用户，显示的内容也不尽相同，显示的内容只是和登陆部门最密切，对他们有用的信息。例如，如果是现场工程部门，那么显示的内容就是以故障原因、排故方法、参考的 FIM、AMM 手册编号为主要

内容。

在结果的局部修改上，是根据检索出来的案例的相似度和一致属性的个数两个因素来评定的，将这两个数据的乘积作为诊断出的主案例，其他案例作为辅助参考案例进行修改。

推理的程序流程如图 4-9 所示。

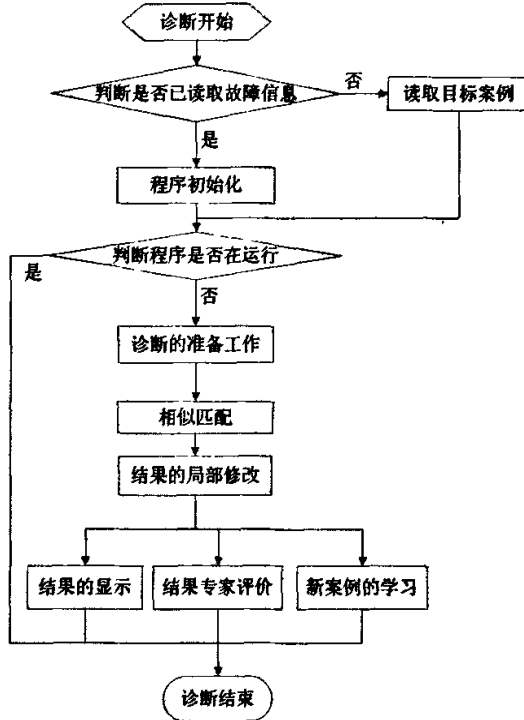


图 4-9 推理的程序流程图

在程序初始化的过程中，将程序中用到的变量，诸如存放已经匹配的目标案例征兆的变量、存放最终匹配出来的案例号的变量、存放属性字段的次序的变量等都需要进行声明并初始化。

诊断的准备工作包括了读取 BIT 信息索引号、读取征兆检索次序、读取案例检索次序、获取案例匹配的字段名次序。这些准备工作都是为相似匹配做准备，将需要的变量都存储到一些可变大小的数组当中，由于它们的数量不同，采用可变大小数组的数据结构可灵活使用，不受存储空间的限制。下面分别就各准备工作的流程做一个简要分析，另外对相似匹配进行详细讲述。

## 4.3.2.1 推理前准备工作

- (1) 读取 BIT 信息索引号。通过上一节介绍的功能模块，程序可以将读取到的 BIT 信息存储到 BIT\_Info 这个 CString 类型的变量中，程序只需要连接数据库，从数据库中的 BIT\_Index 表中读取 BIT\_Info 对应的字段“ID”即可，它的流程图如图 4-10 所示。

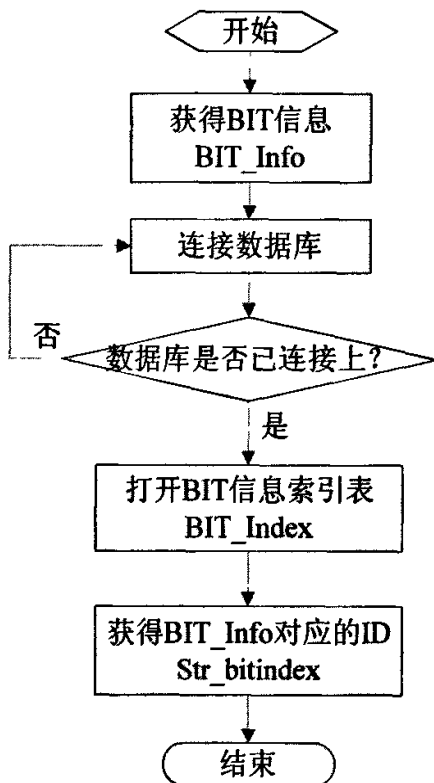


图 4-10 读取 BIT 信息索引号流程图

- (2) 读取征兆检索次序。检索 Fault\_Symp\_Retrieve 表中与读取到的 BIT 信息一致的各个字段的征兆名称，将这些征兆名称按检索的次序存储到 array\_symp\_retrieve\_order 这个 CStringArray 动态字符串数组中。它的流程图如图 4-11 所示。
- (3) 读取案例检索次序。这里以 BIT 信息的索引号为限制检索条件，在 Case\_Retrieve 表中找到与 BIT 信息相对应的所有案例，并且在检索过

程中,按照案例的匹配成功次数降序进行检索,按照降序存储到变量 `array_case_retrieve_order` 中,这个变量是 `CStringArray` 动态字符串数组类型。

图 4-12 是它的工作流程图。

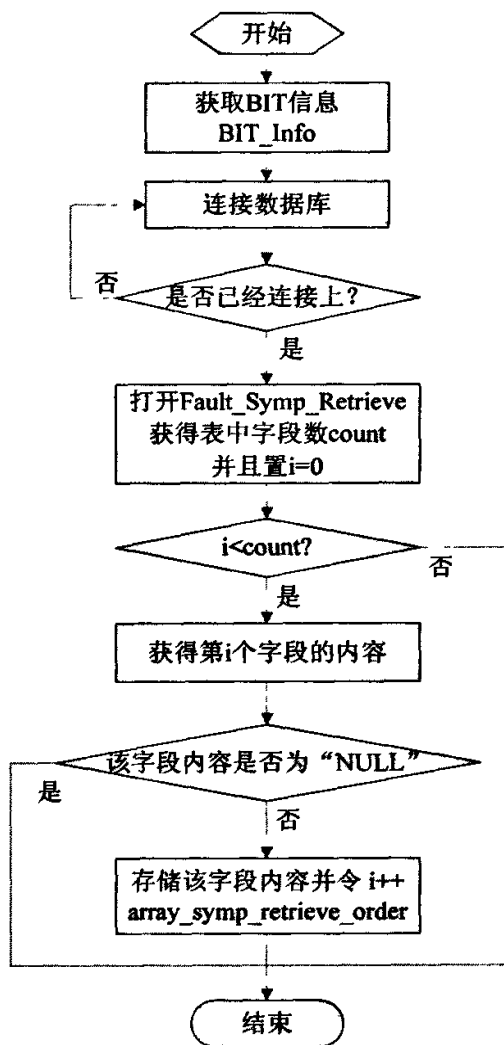


图 4-11 读取征兆检索次序流程图

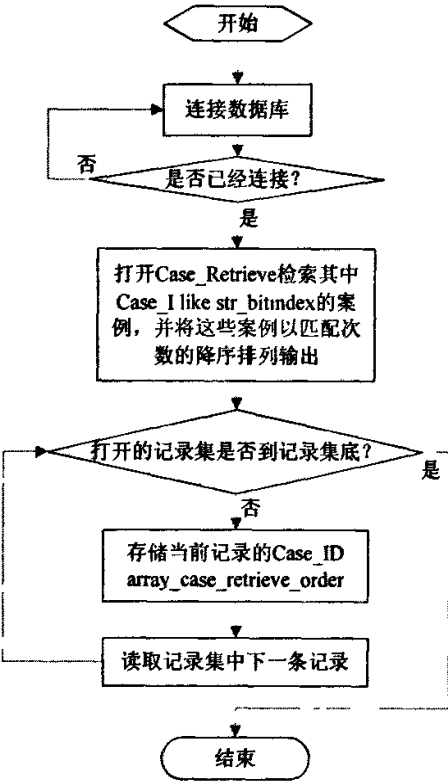


图 4-12 读取案例检索次序流程图

- (4) 获取案例匹配的字段名次序。这里是把前面已经读取出来的故障征兆对应到征兆表中的字段名上，因为检索的故障征兆的顺序是被打乱的，而在征兆表中的征兆又与征兆值表中的征兆顺序相同，所以需要这个过程将征兆与字段名进行映射。可以用图 4-13 表示这种映射关系。

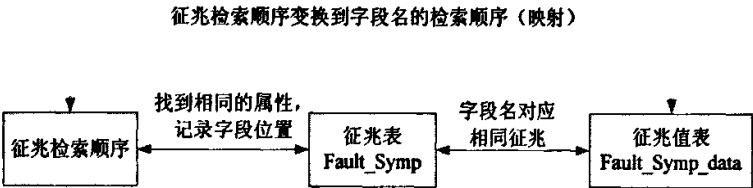


图 4-13 征兆检索顺序到征兆字段名检索顺序的映射

这种映射关系建立的流程图如图 4-14 所示。

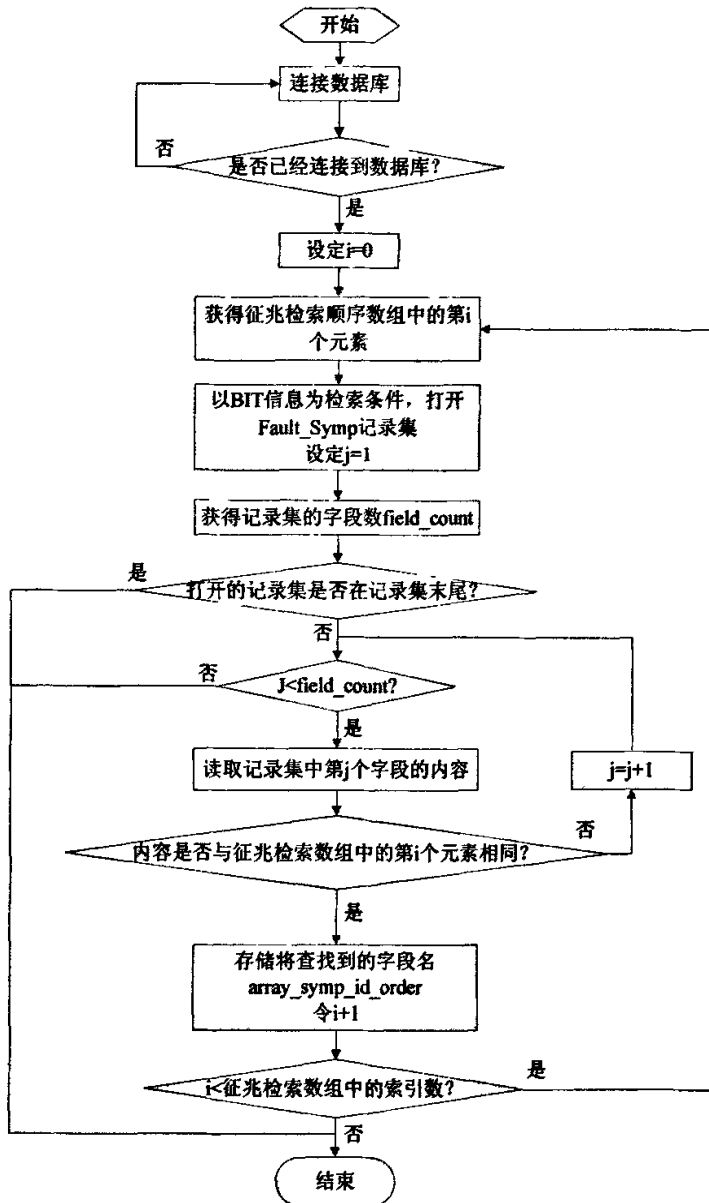


图 4-14 获取案例匹配的字段名次序流程图

以上是对诊断准备工作的介绍,可以看出诊断的准备工作一方面是确定进行征兆匹配的先后顺序,也就是征兆值表中字段的匹配顺序;另外一方面是确定案例进行匹配的先后顺序。这两个顺序列都存储到了相应的数组中,它们的元素索引号就是它们的匹配顺序,在推理操作中,只需要按照序号来进行就可以了,通过这样的准备,为后面的相似匹配减轻了负担,加快了匹配的速度。下面就对相似匹配做详细介绍。

#### 4.3.2.2 案例相似匹配

这个模块是整个系统的核心，是前述的算法的具体实现。在这个模块中要涉及到征兆权值表、征兆值表和临时表案例相似度表的操作。

首先获得待匹配的历史案例（源案例），然后获得征兆的匹配次序。首先进行征兆匹配，然后计算整个案例相似度。最后将匹配出的案例加入到临时表中。

这个匹配过程的流程图如图 4-15 所示。

从图 4-15 可以看到，推理过程是采用案例套属性这种方式进行的，也就是一个案例下的属性进行匹配，当所有属性都匹配成功，则计算这个案例的相似度，同时记录这个案例中与目标案例之间征兆值完全一样的个数。在这个过程中，目标案例的征兆和征兆值一旦参加过匹配，则就会从存储这个征兆的变量中删去，这样在案例匹配完成后，如果这个变量中仍然还有征兆未匹配，则这几个征兆会留到案例的学习和案例的修改中去使用。最终诊断的结果全部在临时表 Case\_similar 表中存储，程序只需要读取其中的 Case\_ID 就能从案例表中检索出相应的案例信息。对于案例相似匹配之后的收尾工作就是要将匹配出来的案例在 Case\_Retrieve 中对应的匹配成功次数加 1。

#### 4.3.3 案例的学习程序模块

案例的学习是不断的填充历史案例库，提高诊断的准确性，但也不可回避的就是增加了案例检索的时间，这是两个互为矛盾的问题，需要从中找到平衡点。

系统中设计了 3 种案例学习的情况（不包括专家评价或修改后再学习）：

（1）案例库中不存在待诊断的故障的 BIT 信息。



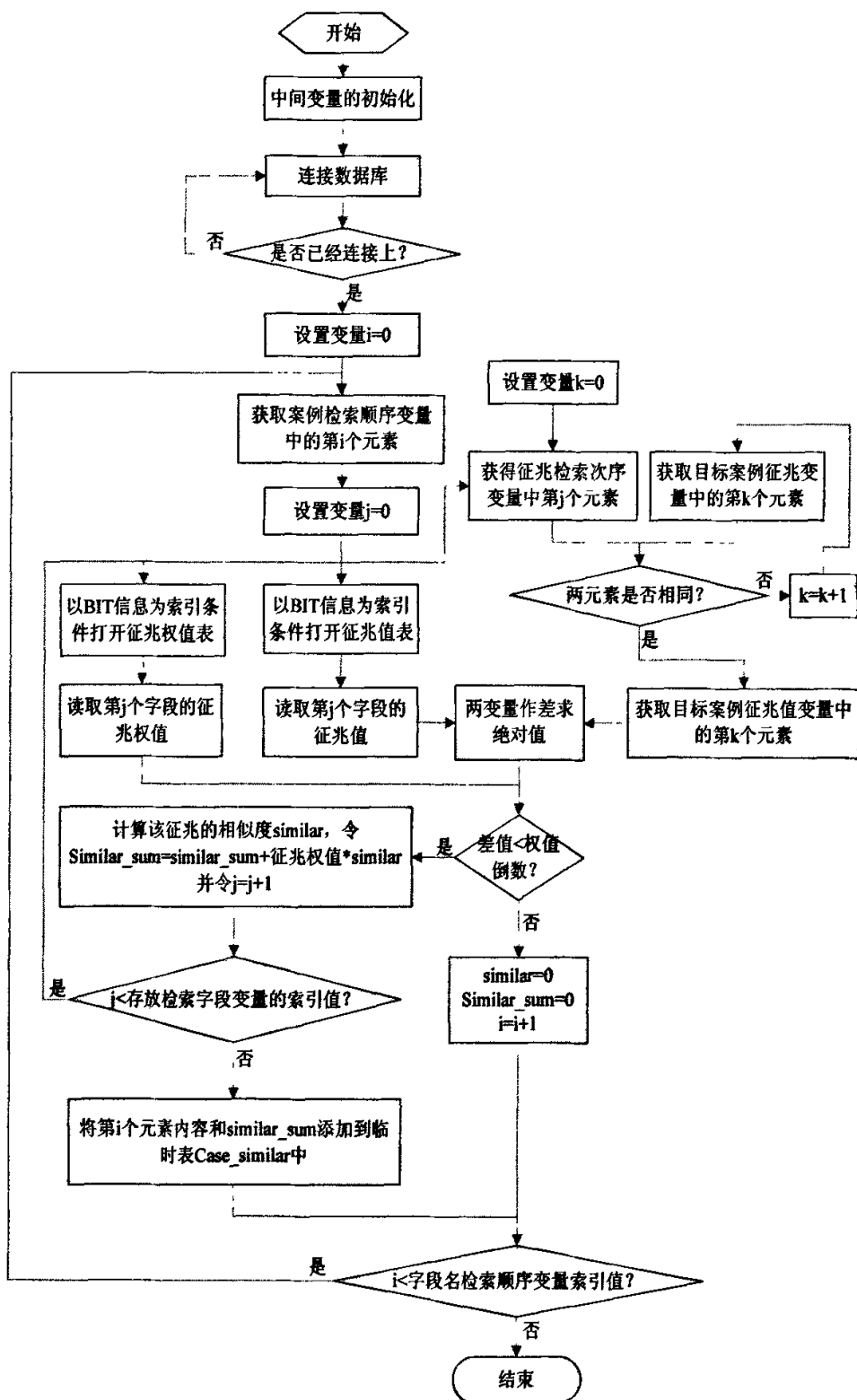


图 4-15 案例相似匹配流程图

(2) 虽然历史案例库中存在待诊断的 BIT 信息, 但是没有相匹配的案例。

(3) 目标案例中的部分或全部征兆不同于案例库中的征兆。

案例的学习模块会涉及到 BIT\_Index、Fault\_Symp、Fault\_Symp\_Imp、Fault\_Symp\_Val、History\_Case、Fault\_Symp\_Retrieve、Case\_Retrieve 这些表的操作。

下面就将对这几种学习模式作简要分析。

#### 4.3.3.1 新的 BIT 信息

新的 BIT 信息是指在 BIT\_Index 中没有找到和目标案例中的 BIT 信息一致的, 称为新的 BIT 信息, 这时候所进行的操作实际上是在各个表中添加内容即可, 无需对现有内容进行更改。这种情况下的程序流程图如图 4-16 所示。

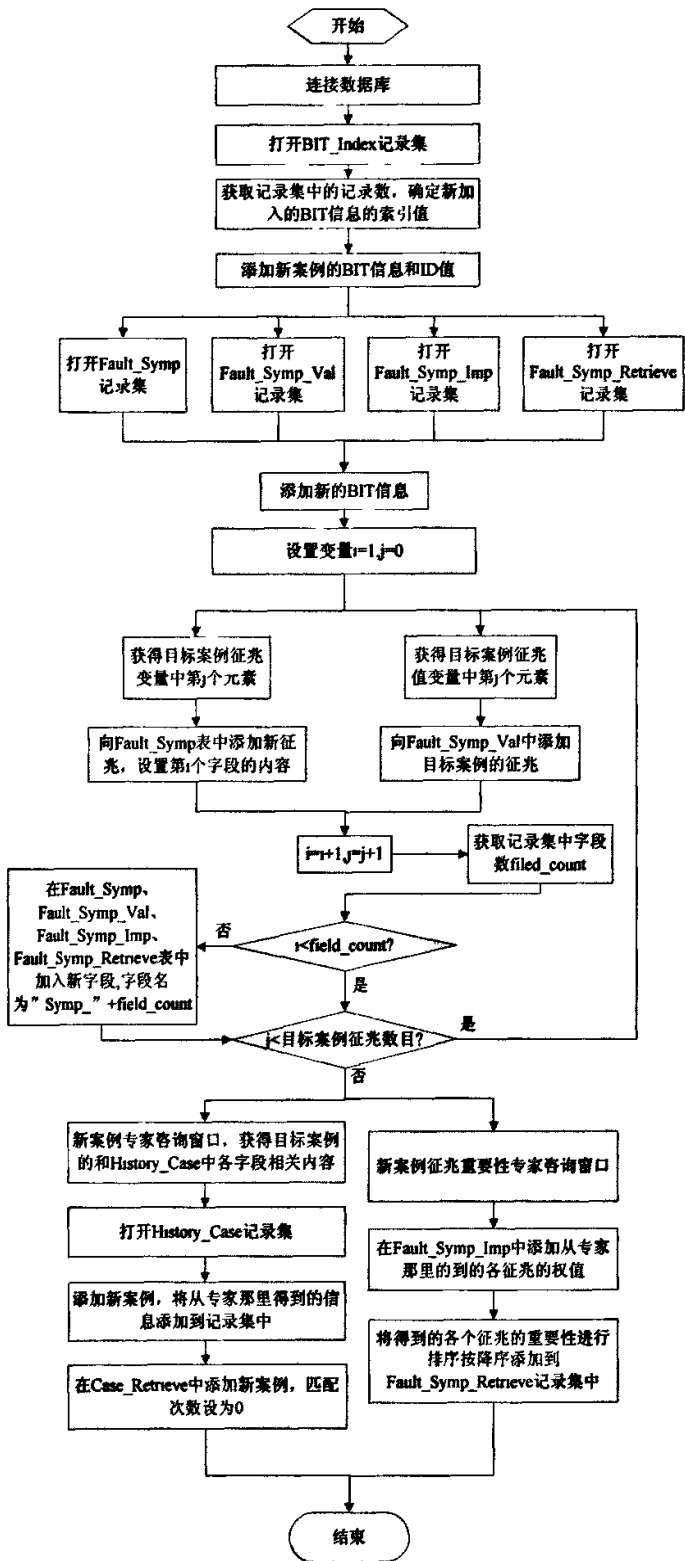


图 4-16 新的 BIT 信息案例学习流程图

## 4.3.3.2 无相似案例可匹配或是有多余故障征兆

虽然 BIT\_Index 表中存在目标案例的 BIT 信息索引,但是历史案例库中没有可以匹配相似的案例,这种情况下的学习流程如图 4-17 所示。

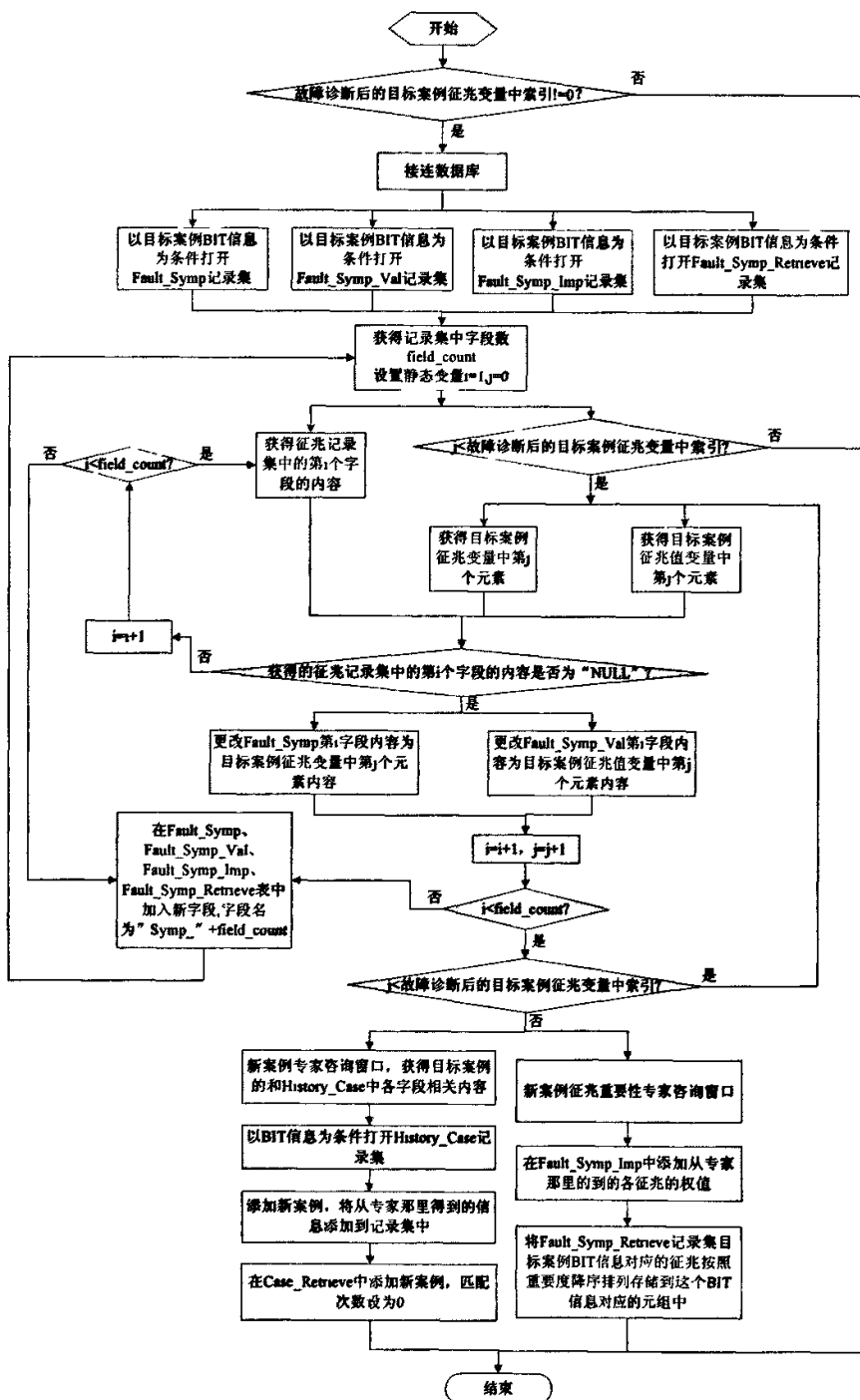


图 4-17 无相似案例匹配的学习流程图

## 4.3.3.3 征兆无差异，但无相似案例

上述的流程是在有些目标案例的故障征兆不包含在历史案例故障征兆的范围之内的情况，当目标案例故障征兆全都包含在历史案例的故障征兆之内时，案例学习的条件就是最终没有得到匹配的结果，这时的学习流程如图 4-18 所示。

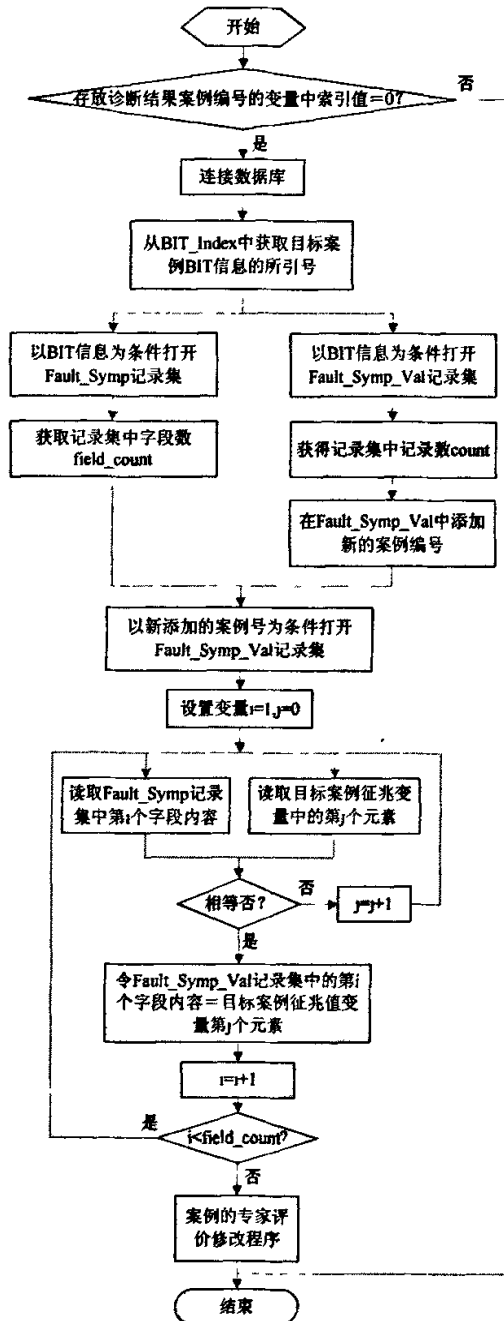


图 4-18 学习流程图

#### 4.3.4 案例的专家评价程序模块

如果用户对于诊断的结果不满意,可以在诊断结果界面点击评估修改按钮,进入评估修改界面,提出正确的意见以供系统学习,补充案例库。这个评价的过程是系统与专家互动的过程,需要专家将领域内能够解决目前故障的正确知识表达出来。评价的最终目的是将评价后的知识添加到历史案例库的相关字段中。它一般是在案例的学习基础上进行的。系统将会对评价后的案例进行再学习。学习的过程可以在 4.3.3 中进行。

#### 4.3.5 数据库的案例维护程序模块

这里所进行的维护就是利用粗糙集理论计算历史案例库中征兆的权值。是按照 BIT 信息来进行维护的,每一个 BIT 信息都对应了很多历史案例,这里就是要从征兆值表中将这些案例的每个征兆对应的征兆值进行处理。因为对于同一种 BIT 信息,表中每个字段对应的征兆种类是一样的,所以可以进行这种操作。这里所用到的算法或是方法都是在前述章节中介绍过的,这里就不再重复。

整个维护要经历以下三个阶段:

- (1) 征兆数据的离散化;
- (2) 征兆的重要性计算;
- (3) 征兆检索树的重新建立;

确定一个 BIT 信息,然后打开这个 BIT 信息对应的所有案例的征兆值表记录集。离散化的顺序是按照每个征兆进行离散化,然后离散化其他征兆。每离散化后一类征兆就将离散化后的数值写入离散数值表中。当所有的征兆全部离散化之后就进行重要性的计算,重要性计算完后进行征兆检索树的建立,建立完后清空该离散数值表。然后再确定下一个 BIT 信息,往复进行上面的处理过程。

在征兆重要性计算时,首先将各个案例的排故方式数字化后写入征兆离散表的 DONE 字段中,这里定义的排故方式采用从 FIM 中掌握的普遍的 3 种类型作参考,它们是故障保留、故障件更换、故障件修理,它们数值化后的值分别为 0、1、2。这样,在征兆数据离散化之后,征兆离散表就称为一个决策表,这个决策表的论域  $U$  就是案例编号,它的大小由案例的个数决定,条件属性  $C$  就是各个征兆组

成的集合, 决策属性  $D$  就是排故方式。这样粗糙集理论中的依赖度计算就可以基于这个表来进行了。

征兆值离散化的流程图如图 4-19 所示。

重要性计算的流程图如图 4-20 所示。

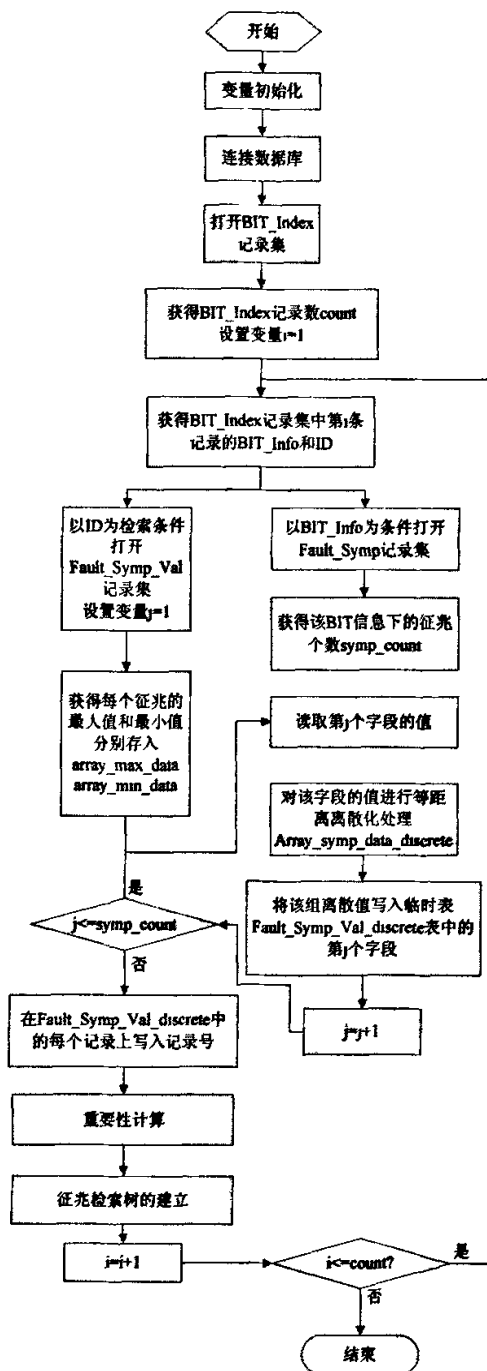


图 4-19 征兆值离散化流程图

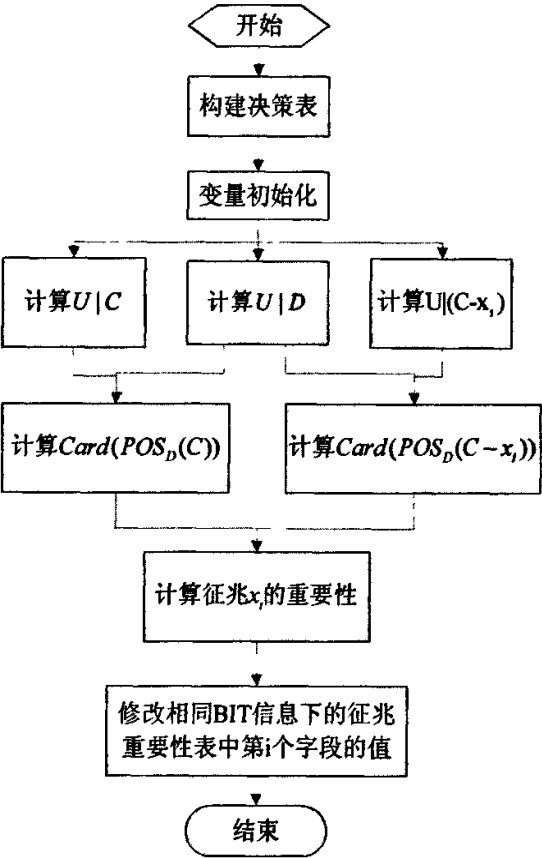


图 4-20 征兆重要性计算流程图

征兆检索树的建立是利用征兆重要性计算结束后修改的故障征兆权值表中的权值来完成的。首先将每个征兆的权值读入内存，然后将这些值降序排列，将排序的征兆值对应的征兆名称写入故障征兆检索表中。这一过程的流程图如图 4-21 所示。

4.3.6 故障诊断结果的保存与分析程序模块

故障结果的保存采用将诊断出的案例对应的历史案例库中的信息存储到一个文本文件中。以诊断时间，故障飞机型号作为文本文件的名称，这样可以利于历史备案。对于实时的给用户显示的内容则是通过一定的修改显示在程序运行界面的。修改的过程，我们是通过了一定的策略来完成的。首先将案例的相似度和它



的一致征兆数两个值作乘积得到的数值作为案例主次的参考值，这个值最大的作为诊断的主案例，而对于其他的案例作为次要的案例，通过比较各个案例中所涉及到的 FIM\_NO 和 AMM\_NO 来判断需要在主案例中的排查方法、故障原因、故障部件字段中增加什么样的内容。如果比较的过程中发现案例中出现的 FIM\_NO 或是 AMM\_NO 不同，则要记录不同的 FIM\_NO 或是 AMM\_NO 的内容，并检索所有匹配出的案例中这种不同的 FIM\_NO 或是 AMM\_NO 总共出现的次数，以出现次数最多的前 2 个作为修改的最终参考，来修改主案例中的相关内容，最终作为故障诊断信息供用户直观的参考。

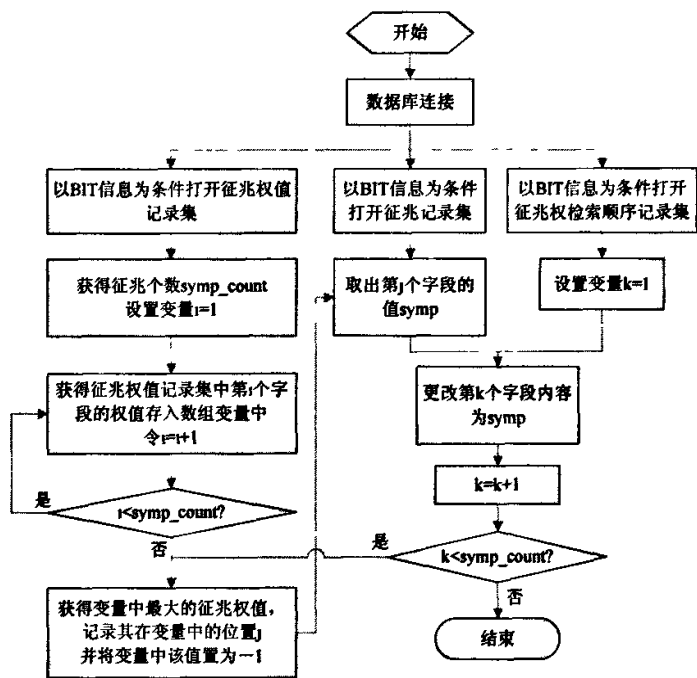


图 4-21 征兆检索树建立流程图

4.4 本章小节

本章主要是对故障诊断专家系统的设计进行研究，按照数据库的设计规范和流程设计了该专家系统的案例知识库。根据系统的设计目标，设计了系统功能模块。对系统中的各个模块进行了详细的分析和设计，清晰的给出了模块的流程图，对模块的详细功能进行了详细的介绍。

## 第五章 故障诊断专家系统的运行与验证

系统的运行需要友好的用户界面，能够令用户了解系统的运行情况，另一方面也是更为重要的方面就是系统能够正确的运行，正确运行的表现就是系统能够准确的提供诊断结果或是能够提供最有参考价值的诊断意见。

### 5.1 用户交互界面设计

#### 5.1.1 用户登陆界面

系统运行，首先需要用户连接到数据库，这里采用独立的用户登陆界面来处理。在结果输出上系统会根据不同的用户群显示故障诊断结果。如图 5-1 是用户登录界面，用户名只是区分不同的用户群组而确定的，不同的用户可能属于不同的用户群组中，这些只需要一个用户名即可。如果没有用户名，可以通过点击注册新用户和注册临时用户，结果一经显示，这个用户即刻销毁。如图 5-2 是临时用户注册界面。

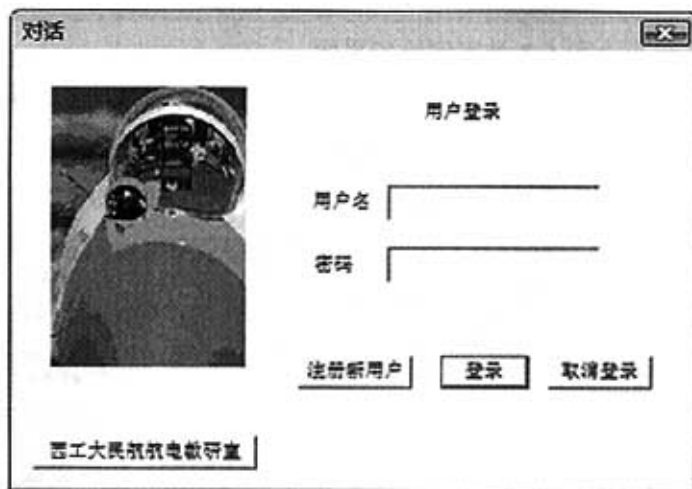


图 5-1 用户登录界面

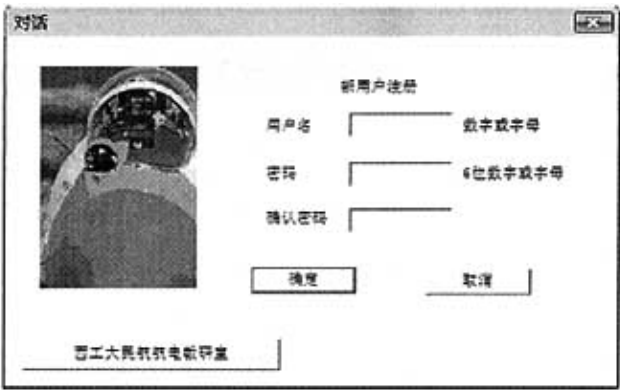


图 5-2 新用户注册界面

5.1.2 历史案例维护界面

系统提供了两种类型的案例维护方式，一种是在案例学习后系统自动判断是否需要进行案例的维护，另一种是由用户自定义进行的。第一种自动的方式没有提供与用户交互的功能，这里所介绍的对话框是第二种方式下的交互界面，提供给用户自定义进行历史案例库中征兆的权值维护，用户可以选择对哪种类型的 BIT 信息对应的案例进行维护，可以选择多种 BIT 信息，也可选择对所有案例进行维护。系统在给用户提供的可选 BIT 信息中，可以有两种选择模式，一种是经过过滤的，就是经过系统判断可以不需要进行维护的案例，这些案例将不会出现在用户选择的范围内；一种是将全部 BIT 信息提供给用户，由用户自己作决定到底选择哪些 BIT 信息。图 5-3 是案例维护界面。

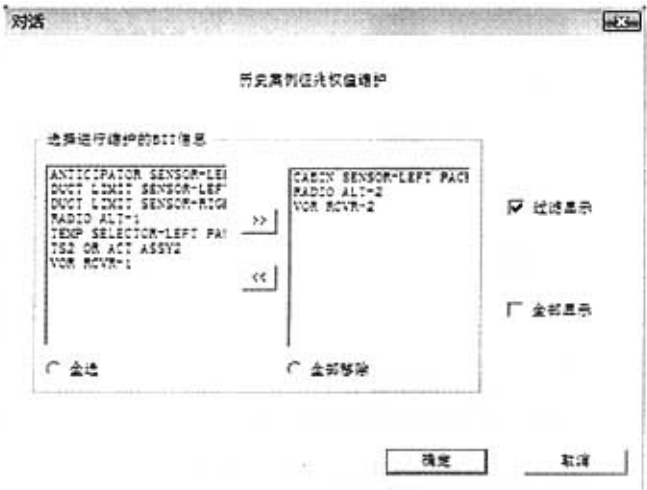


图 5-3 自定义案例维护界面

5.1.3 诊断结果的评价界面

当故障诊断结束后，用户可以根据实际情况针对系统诊断结果进行评价处理，实际上是对案例的诊断结果进行修改，协助系统对新案例的学习。这个过程是可视化的和用户或专家互动来进行的。专家或用户直接对诊断结果进行修改即可。无需其他复杂的操作。图 5-4 中的“查看故障征兆信息”按钮点击后将会显示该案例对应下的故障征兆及各个征兆的具体取值。

图 5-4 故障诊断评价表单

5.1.4 新案例专家咨询界面

当目标案例中的故障 BIT 信息在历史案例库中检索不存在时，系统会转入新案例学习模式，此时会显示给用户如图 5-5 的所示的“新故障案例报告单”，该表单将向用户咨询新案例的处理方式以及故障原因等一系列问题，用户只需要填写该表单即可。在填写完该表单后，点击确定系统还会提示用户是否需要输入各故障征兆的权值，如果用户不输入，则系统会自动为这些故障征兆填写默认的权值，如果用户需要填写，将会提供给用户如图 5-6 所示的“故障征兆权值咨询表单”供用户填写。采用单个征兆显示的方式，如果有多个征兆就会多次显示该窗口供用

户分别输入征兆权值，直到全部征兆处理完后结束。

对话框

新故障案例诊断报告单

BIT信息

故障原因:

故障方法:

FIX\_NO:

AXOM\_NO:

故障LNU:

查看故障征兆信息

故障处理方式  
☐ 保留  
☐ 换件  
☐ 维修

确定 取消

图 5-5 新案例的诊断结果报告单

对话框

请输入该征兆的重要性(级权值)

BIT信息:

征兆名称:

级权权值:

确定 取消

图 5-6 新案例故障征兆专家咨询单

从图上的表现形式看似图 5-4 与图 5-5 表单结构是一样的，但是在实际系统运行中，图 5-4 中各个编辑框中和可选择框中会显示该案例诊断的结果。

5.2 专家系统运行的结果验证

5.2.1 专家系统运行情况验证

为了检验系统的工作情况，准备各种不同情况的目标案例来验证系统运行的效果。

5.2.1.1 目标案例1（单 BIT 信息，征兆与历史案例库中案例征兆一致）

经过运行故障信息读取程序读到的故障信息如表 5-1 所示。

表 5-1 目标案例 1 信息表

BIT 信息	ANTICIPATOR SENSOR –LEFT PACK –NO GO			
征兆项	Cabin temperature controller BITE	Pin1-14	Pin2-15	Pin3-16
征兆值	1	0	1	0

如果用户未读入该故障信息，系统经过判断会提示用户点击“读取故障信息”按钮，进行故障信息的读取。接下来用户单击“故障诊断”，系统开始运行故障诊断程序最终显示结果如图 5-7 所示。

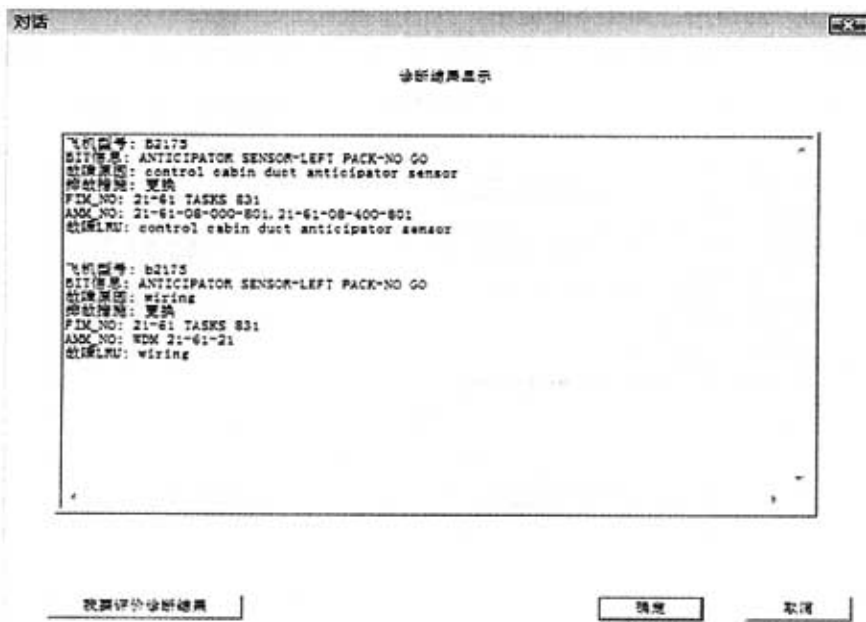


图 5-7 故障诊断结果显示界面

诊断结果表明案例库中存在两条历史案例与目标案例 1 匹配成功。这种情况下系统没有对结果进行修改，因为目标案例 1 中的故障征兆和历史案例中的征兆一致，并且都为定性数据数值化，在 3.7 节中的验证表明，这种情况下无需进行修改。只是用户需要进一步评价而已。

5.2.1.2 目标案例 2（单 BIT 信息，征兆有部分不同）

经过运行故障信息读取程序读到的故障信息如表 5-2 所示。

表 5-2 目标案例 1 信息表

BIT 信息	TEMP SELECTOR-LEFT PACK-NO GO					
征兆项	Cabin temperature controller BITE	Pin1-19	Pin2-20	Pin4-10	Pin5-15	Pin6-21
征兆值	1	1	0	0	0	1

该目标案例中存在的某些征兆项是历史案例库中不存在。所以会在诊断结果结尾有“目标案例有新征兆项未匹配，系统自动进行了学习。”这样提示性的语句，起到通知用户的作用。如图 5-8 所示。

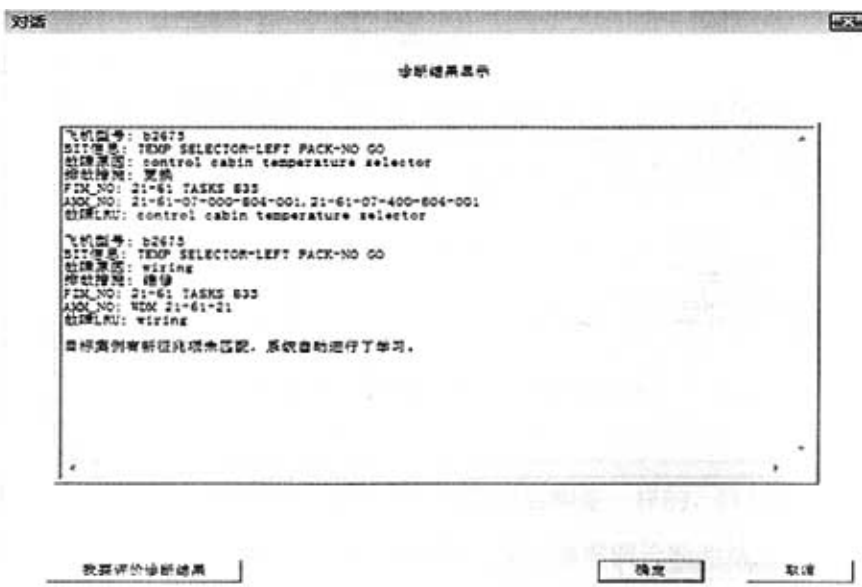


图 5-8 故障诊断结果显示界面

5.2.1.3 目标案例3（多 BIT 信息）

经过运行故障信息读取程序读到的故障信息如表 5-3 所示。

表 5-3  目标案例 3 信息表

BIT 信息	TS2 OR ACT ASSY2				
BIT 信息	VOR RCVR-1				
征兆项	2# 油门杆	扭力电门	自动油门	DFCS BITE	VOR/Marker Beacon Receiver BITE
征兆值	0	1	1	1	1
BIT 信息	TS2 OR ACT ASSY2				
BIT 信息	VOR RCVR-1				
征兆项	Channel A	PinH8-A4		PinG8-B4	
征兆值	0	0		0	

系统分别对这两种 BIT 信息进行诊断，然后将诊断的结果显示给用户。如图 5-9 所示。

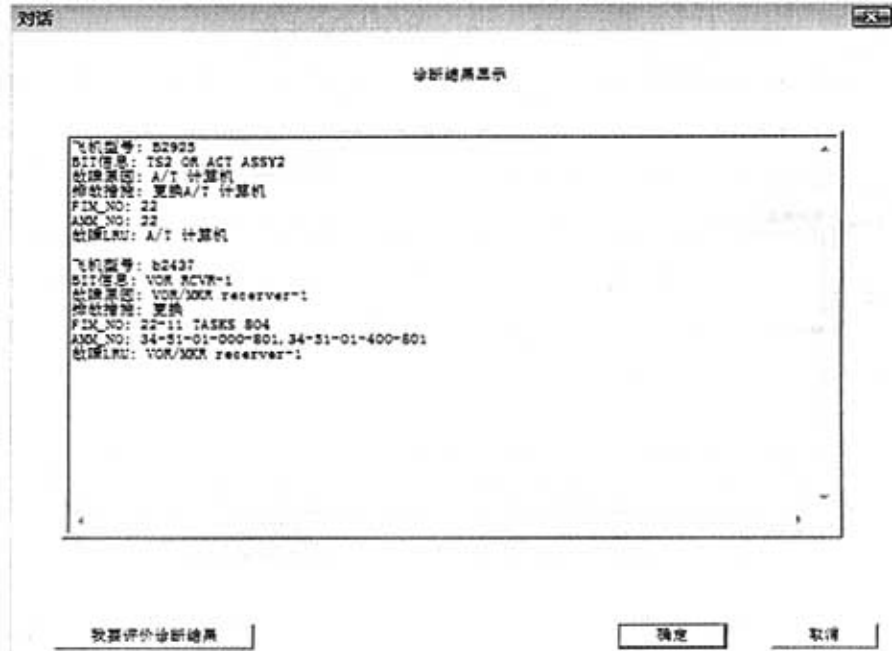


图 5-9 故障诊断结果显示界面



5.2.1.4 目标案例4（新 BIT 信息，历史案例库中没有该 BIT 信息）

经过运行故障信息读取程序读到的故障信息如表 5-4 所示。

表 5-4  目标案例 4 信息表

BIT 信息	KEY INNERLOCK			
征兆项	HF communication System BITE	Coaxial Cable inspection	P18-2	P6-1
征兆值	1	0	0	0

由于这种 BIT 信息在历史案例库中不存在,所以系统会提示用户“诊断失败!”,同时会弹出案例专家咨询报表需要专家填写,以完善该条案例的系统学习。图 5-10~5-12 是这个过程的图形界面。

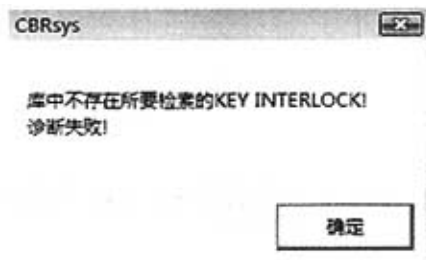


图 5-10 提示用户诊断失败

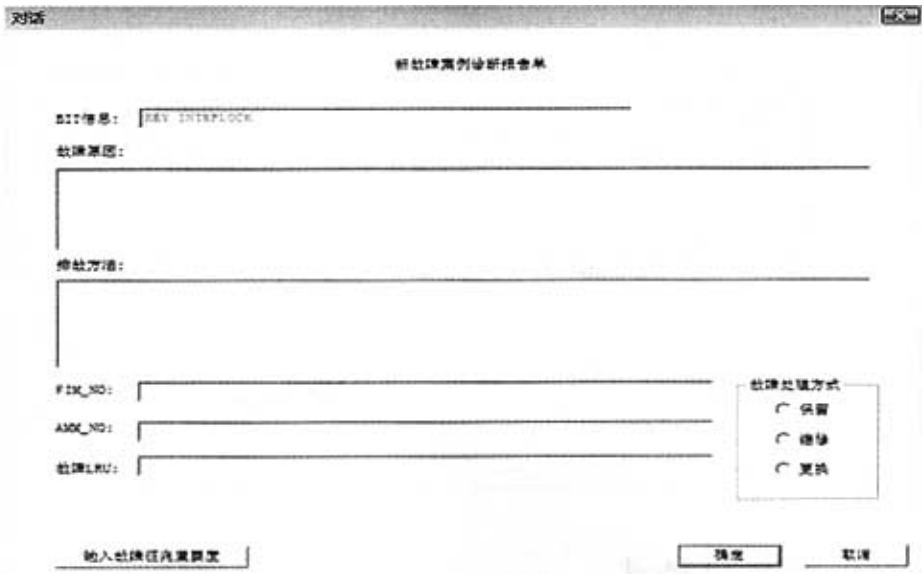


图 5-11 新案例咨询界面



图 5-12 新征兆经验重要度咨询界面

在 5-12 的界面中如果用户未填入经验权值，则系统会自动给该征兆添加一个默认权值。

5.2.1.5 目标案例 5（普通案例 1）

这里仍然使用在 3.7 节中所使用的案例作为目标案例来检验系统的运行效果。该目标案例的 BIT 信息和征兆信息如表 5-5 所示。

表 5-5 目标案例 5 信息表

BIT 信息		RADIO ALT-1					
征 兆 项	DFCS BITE	LRRA BITE	channel A	channel B	radio transmeter-1	pinB4-B2	pinA4-B3
征 兆 值	1	0	0	0	1	1	0

由于在上面几个章节中已经介绍了所有相关用户界面的作用及其如何操作，这里就不再累述。只将案例诊断结果以表格的形式表示如下（表 5-6）：

表 5-6 案例诊断结果一览表

项	飞机型号	BIT 信息	故障原因	FIM_NO	AMM_NO	故障 LRU	排故措施
内容	B2867	RADIO ALT-1	radio traceiver s-1 发生 故障, 板上的 pinB4-pi nB2 有 故障。	22-11 TASK 805	22-11-3 3-000-8 01 22-11-3 3-400-8 01 WDM 34-33-1 1	radio traceiver s-1; wiring	更换故障 LRU, 维修板上 pinB4-pi nB2 之 间的线路

5.2.1.6 目标案例 6（普通案例 2）

使用 3.7 节中的一些案例作为目标案例，如表 5-7 所示。

表 5-7 目标案例 6 信息表

BIT 信息			STANDBY ALTIMITER PROBLEM	
征兆项	继电开关	马赫数	飞行高度	高度的差值
征兆值	1	0.473	10032	430

该目标案例的诊断结果如表 5-8 所示。

表 5-8 案例诊断结果一览表

项	飞机型号	BIT 信息	故障原因	FIM_NO	AMM_NO	故障 LRU	排故措施
内容	B2232	STANDBY ALTIMITER PROBLEM	虚 假 BIT 信 息	/	/	/	/

	B2657	STANDBY ALTIMITER PROBLEM	皮托管 有裂缝			皮托管	更 换 皮 托 管
	B2148	STANDBY ALTIMITER PROBLEM	皮托管 被破坏			皮托管	更 换 皮 托 管

5.2.2 专家系统运行结果综合分析

通过 5.2 节的几个案例的运行情况，我们可以看到针对不同情况下的案例该专家系统都会按照预计的程序来处理。从目标案例 1 开始分析，这个案例的征兆项与历史案例库中的征兆项完全相同，在这种情况下，系统通过检索历史案例库中与目标案例 BIT 信息相关的历史案例来得到诊断结果。可以从结果中看到，系统得到了两个历史案例，通过与该故障的实际排故结果对比来看，能够给与维修人员参考的作用。下面分析目标案例 2，由于这个案例中的部分征兆项在历史案例库中是不存在的，所以对于系统来说是一个相对陌生的案例，系统只能通过与目标案例征兆项相同的进行诊断，得到了最终的结果虽然与真实情况有些偏差，但是对于系统推理的本身是不存在问题的，因为系统在得到诊断结果之后，会自动的提示刚才的故障目标案例部分征兆未匹配，系统进行了学习，有了这样的过程，就能使系统自我丰富历史案例库，提高今后诊断的准确性。同时还要求用户给与该诊断案例的评价意见，这样便丰富了系统学习的手段。对于目标案例 3 这种多 BIT 信息的组合故障的情况，系统采用了分别对各个 BIT 信息进行诊断的方法，最后将各个诊断的结果综合显示，通过诊断，与实际情况一致，可以较为准确的指导用户进行维修。对于目标案例 4，系统的历史案例库就根本不存在与此 BIT 信息对应的 BIT 索引，系统采用了另一种学习的方式，就是要求用户输入该案例的每个征兆值的重要性，对于用户没有输入的情况下系统采用默认的值给与填充，同时要求用户根据实际情况填写新故障案例诊断报告单，最终将该新案例添加到历史案例库中。对于目标案例 5 和目标案例 6，可以将它们分属到两种类型中去，由于历史案例的获取瓶颈问题，这里的历史案例库中收集到的大部分故障征兆项

对应的征兆值都是定性的，系统只是采用了将定性的数据数值化后进行故障诊断，而那些具有连续属性的征兆值比较少，这里目标案例 5 就是较为普遍的定性的征兆项，而目标案例 6 就是具有连续属性的征兆项。可以从这里看到，系统在处理如同目标案例 6 这种案例时会给出不只一个诊断结果，一般是 3 个以上，这种情况是系统设计时允许的，由于连续属性的数值不像定性属性那样是属于精确匹配的，连续属性的征兆的匹配在前面的章节中已经讲到是满足一定约束条件下的匹配，所以对于结果中包含多个案例的情况是允许的。

可以通过上述对各个目标案例的诊断结果的总结看到，系统在进行案例诊断过程中，考虑到了各方面的情况，包括案例的评价，案例的修改，案例的学习，而且还能人性化的提示用户系统下一步需要进行的操作，能够使用户很容易的上手操作。而对于系统诊断出的结果，它受到多方面的因素的限制，包括历史案例库中案例的数量、历史案例收集的能力等，但是就从该系统的推理能力来看，还是能够较为准确的提供给用户诊断结果，帮助用户进行飞机的故障诊断。

通过系统对于实例的运行情况，可以从整体上对系统有如下的评价：

- (1) 该专家系统满足了设计之初的设计需求，能够快速的检索出与目标案例最相匹配的历史案例，即诊断结果，诊断结果能够满足机务人员或是排故人员的工作需要。
- (2) 系统提供了自我完善的学习功能，从而系统会随着时间的推移不断丰富其知识库中的内容，为更加准确的处理问题提供了坚实的基础。
- (3) 系统为用户提供了交互的窗口，用户可以实时更改系统中的案例信息，虽然这是一种被动的学习方式，但是对于系统而言是很好的弥补手段，都是在为下一次能够精确的给出诊断结果做准备。

### 5.3 本章小节

本章通过实际运行所设计的专家系统，运行不同情况下的诊断案例，表明系统运行正常，能够满足系统的设计目标。

## 第六章 总结与展望

### 6.1 本论文工作的总结

论文首先分析了机载设备故障诊断的现状，专家系统与故障诊断的研究现状与发展方向，然后简要介绍了专家系统的组成结构。然后研究给出了基于 CBR 的专家诊断模型，深入研究了 CBR 系统中属性权值的确定方法以及检索匹配的算法问题，最后设计实现了一个诊断专家系统。系统能够完成 CBR 系统流程的基本功能，实现了故障诊断的要求，满足了系统设计的目标。

#### 6.1.1 本论文的主要成果

1、针对目前机载系统的故障诊断的特点，研究了基于飞机实时的 BIT 信息、飞行参数、FDE 等故障征兆完成飞机地面故障诊断的 CBR 方法和技术。

2、提出了基于 CBR 的飞机故障案例的框架的表示法。该方法能够准确的表示一个机载设备历史上故障的每次诊断经验。并且采用了粗糙集的方法对故障案例对应的故障征兆进行简约处理，同时针对征兆值计算该征兆值所表示的 BIT 信息的重要度。

3、利用征兆的重要度组织征兆的检索匹配策略，重要度越大的征兆先匹配，以征兆重要性的倒数作为匹配约束条件，建立了局部匹配的过滤策略。在满足局部匹配约束的条件下，利用最近邻算法计算案例的全局相似度。

4、历史案例也按照案例历史上被成功匹配的次数作为检索的条件，次数越多的则最先检索进行匹配，最小的最后检索。综合 3 和 4 的检索方法，提出了一种推理算法，该算法经过验证能够快速准确的得到诊断结果。

5、设计实现了一个基于 CBR 的飞机实时故障地面诊断系统，系统能够完成 CBR 系统的基本功能，实现了故障诊断的要求，满足了系统设计的目标。

### 6.1.2 现有成果的局限

尽管提出了一些模型、算法和机制，在验证算法机制的过程中，由于受到案例获取困难，条件的限制，为能够实现海量案例数据的测试与评估，文中只是小范围的进行了验证。因此模型、算法中存在的问题还有可能没有完全的表现出来。由于机载设备的关联性较大，对于一种 BIT 信息对应故障征兆的完全性与准确性还需要更多时间，更多案例来证明。

由于所收集到的案例的征兆大部分都是定性的数据，这样就不能够充分证明在粗糙集一节中所用的离散化方法的有效性，是否非常适合将该方法用于这里会随着案例库中案例的不断增加而慢慢明确。

## 6.2 展望

本文中是针对机型来进行诊断的，在系统的不断完善过程中，能够做到以每架飞机作为监视主体，针对不同的飞机（即使它们的型号相同）也可能因为个体存在的差异性造成故障的不同。

积累大量的飞机飞行数据，从数据中寻找飞行参数与故障的关联性。

继续寻求更有效的案例修改策略和评价方法，做到自动与手动并行。

寻求更有效的数据离散化方法，应用到该系统中，提高征兆重要度的代表性。

完善数据库中案例的维护方法，剔除冗余的历史案例，优化数据库中各个表的结构，进一步提高系统运行的效率。

## 参考文献

- [1] 吴明强, 史慧, 朱晓华, 肖开清. 故障诊断专家系统研究的现状与展望[J]. 计算机测量与控制, 2005, 13: 1301-1304
- [2] 黄翔, 黄磊, 刘建, 周绍骑. 现代故障诊断技术研究与趋势[J]. 后勤工程学院学报, 2006, 2: 94-98
- [3] 杜鑫, 唐大全, 杨应成. 机载设备自动测试与故障诊断技术的发展[J]. 计算机测量与控制, 2003, 11: 5-8
- [4] 史彦斌, 段哲民. 航空电子综合测试系统的发展现状及趋势[J]. 计测技术, 2005, 25(4): 1-2
- [5] 吴卫玲. 机载设备智能故障诊断系统的研究[D]: [硕士学位论文]. 南京: 南京理工大学, 2004
- [6] 张荣梅. 智能决策支持系统研究开发及应用[M]. 北京: 冶金工业出版社, 2003.1
- [7] 李永平. 基于 CBR 方法的飞机排故支持技术研究与应用[D]: [硕士学位论文]. 南京: 南京航空航天大学, 2006
- [8] Barletta B. An Introduction to Case-Based Reasoning[J]. AI Expert, 1991, 8(1):43~49
- [9] David B Leake. Case-Based Reasoning-Experience, Lessons, Future Directions[M]. AAAI Press/MIT Press, 1983
- [10] 张仰森. 人工智能原理与应用[M]. 北京: 高等教育出版社, 2004.2
- [11] 李长河. 人工智能及其应用[M]. 北京: 机械工业出版社, 2006.5
- [12] Jocelyn San Pedro, Frada Burstein, Alan Sharp. A case-based fuzzy multicriteria decision support model for tropical cyclone forecasting[J]. European Journal of Operational Research, 2005, 160: 308~324
- [13] 吕琦. 基于知识的民航飞机故障诊断专家系统[D]: [硕士学位论文]. 南京: 南京航空航天大学, 2005
- [14] 吴海桥. 现代大型客机故障诊断专家系统的研究与开发[D]: [博士论文]. 南京: 南京航空航天大学, 2002
- [15] 张祥伟. 现代飞机排故专家系统的设计与实现[D]: [硕士学位论文]. 南京: 南京航空航天大学, 2002



- [16] Felix T.S.Chan. Application of a hybrid case-based reasoning approach in electroplating industry[J]. Expert Systems, 2005, 29: 121~130
- [17] Vargas J.E, Raj S. Developing maintainable expert system using case-based reasoning[J]. Expert System, 1993, 10(4)
- [18] 张文修. 粗糙集理论与方法[M]. 北京: 科学出版社, 2001.7
- [19] 曾黄麟. 粗集理论及其应用[M]. 重庆: 重庆大学出版社, 1996.9
- [20] 李雄飞, 李军. 数据挖掘与知识发现[M]. 北京: 高等教育出版社, 2003.11
- [21] 王国胤. Rough 集理论与知识获取[M]. 西安: 西安交通大学出版社, 2001.5
- [22] 罗春田. BOEING 737 飞机维修经验交流论文汇编[M]. 北京: 中国民航出版社, 1997.9
- [23] 求是科技. Visual C++ 6.0 信息管理系统开发实例导航[M]. 北京: 人民邮电出版社, 2005.4
- [24] 陈联诚, 陈旭东, 区士超. SQL Server 2000 实用教程[M]. 北京: 电子工业出版社, 2005.2
- [25] 刘生平编著. Visual C++.NET 数据库开发技术与实践[M]. 北京: 清华大学出版社, 2005.1
- [26] 航空工业部民机局. 运输类飞机飞行试验手册[M]. 北京: 航空工业部民机局, 1988
- [27] GE Capital Corporation, Boeing portable maintenance[CP] aid-FIM, Rev 27, Jun 10/2005, D633A103-GEF-TD
- [28] 顾德均. 航空电子装备修理理论与技术[M]. 北京: 国防工业出版社, 2001.2
- [29] 张凤鸣, 郑东良, 吕振中. 航空装备科学维修导论[M]. 北京: 国防工业出版社, 2006.2
- [30] 以光衡. 航空机载电子系统与设备[M]. 北京: 北京航空航天大学出版社, 1997
- [31] 何晓薇, 徐亚军. 航空电子设备[M]. 成都: 西南交通大学出版社, 2004.7
- [32] 宋瀚涛, 梁允荣. 关系数据库原理与系统[M]. 北京: 北京理工大学出版社, 1992.7
- [33] 樊尚春. 航空测试系统[M]. 北京: 北京航空航天大学出版社, 2005.7
- [34] 王化祥. 自动检测技术[M]. 北京: 化学工业出版社, 2004.8
- [35] 朱大齐. 航空电子设备故障诊断技术研究[D]: [博士学位]. 南京: 南京航空航天大学, 2002
- [36] 李长奇. an introduction to avionics (for civil aviation)民用航空电子系统概论[M]. 天津: 中国民用航空学院, 1992.6
- [37] David C.Wilson. Case-based Maintenance: The Husbandry of Experience [D].Indiana, USA: Indiana University, 2001
- [38] C.Chiu, N.H.Chiu, C.L.Hsu. Intelligent aircraft maintenance support system using genetic algorithms and case-based reasoning[J]. International Journal of Advanced Manufacture Technology, 2004, 24: 440~446

- [39] B.Interpol. Data Mining and Knowledge Discovery in NBA Data[M]. Boston: Kluwer Academic Publication, 1999
- [40] Jr.W.M.Peck, A.C.Scheraga, P.R.Boisjoly. Assessing the relative efficiency of aircraft maintenance technologies: an application of data envelopment analysis[J]. Transportation Reasearch, Part A (Policy and Practice), 1998, 32A (4): 261~269
- [41] A.V.Skormin,J.L.Popyack. Reliability of flight critical system components and their “history of abuse”[A]. Proceedings of the IEEE 1995 National Aerospace and Electronics Conference, NAECON. 1995, 1: 376~381
- [42] J.Lewitowicz. Models of functional systems’ evolution in diagnostics of airplanes, helicopters and robots[A]. IEEE SMC’99 Conference Proceedings, 1999 IEEE International Conference on Systems, Man, and Cybernetics, 1999, 6: 1006~1008

## 论文发表情况

王波, 宋东, 姜华男, 基于粗集的 CBR 故障诊断案例的检索方法研究, 计算机测量与控制,  $\frac{ISSN1671-4598}{CN11-4762/TP}$ , 2007 年第 11 期

姜华男, 宋东, 王波, FMS 导航制导及系统数字仿真, 电子测量技术,  $\frac{ISSN1002-7300}{CN11-2175/TN}$ , 2007 年第 30 卷第 9 期

## 致谢

随着本论文的顺利完成，富有人生意义的两年半时光即将以一个圆满的句号而告终。在这短暂的时间里，我不仅学到了丰富的知识，更学到了一生受益不尽的生活哲理。所有这些收获是和我的导师宋东副教授的无私奉献分不开的。回顾学习、生活、工作中的点点滴滴，宋老师的严格要求、无微不至的关怀、悉心的指导，给了我莫大的帮助，尤其是宋老师脚踏实地的敬业精神、严谨求实的治学作风、平易近人的高尚品德，更是留给我的最宝贵的记忆。本课题倾注了宋老师大量的心血，论文的顺利完成更是得益于宋老师的耐心教导和精心指导。在此谨向宋老师表示最诚挚的谢意。

在本论文的完成过程中，航电课题组的马存宝副教授给予了我无私的指导和热心的帮助，启发了我的新思路，在此表示衷心的感谢。

感谢朱亚光、姜华男、肖保良、李海浩、陈劲松、周伟六位同学与我进行很多有益的探讨。

感谢系 2004 级研究生班的所有同学，不会忘记一起走过的日子。

最后我要把这篇文献给我的家人，是他们的默默支持，才使我顺利的完成了这篇论文。