

基于 CCSDS MO 服务参考模型消息抽象层设计

刘 宇^{1,2,3}, 刘玉荣²

- (1. 中国科学院 光电研究院, 北京 100094;
2. 中国科学院 空间应用工程与技术中心, 北京 100094;
3. 中国科学院 研究生院, 北京 100049)

摘 要:随着空间技术的高速发展,任务运营管理系统日益庞大、复杂,管理成本也不断增加。针对这种情况,CCSDS 提出了面向服务体系架构的任务运营管理系统参考模型,即 MO 服务参考模型。MO 参考模型意在使任务运营系统成为由多个小型的、标准化的组件组装而成的系统。由于消息抽象层(Message Abstract Layer, MAL)的设计是 MO 参考模型的核心。文中以我国有效载荷运控系统的需求为出发点,同时依据 MO MAL 为参考,结合面向对象方法学,设计模式等理念,对 MAL 进行设计。

关键词:任务运营管理系统;消息抽象层;SOA;设计模式

中图分类号:TP399

文献标识码:A

文章编号:1673-629X(2013)08-0009-05

doi:10.3969/j.issn.1673-629X.2013.08.003

Design of Message Abstract Layer Based on CCSDS MO Service Reference Model

LIU Yu^{1,2,3}, LIU Yu-rong²

- (1. Academy of Opto-electronics, China Academy of Sciences, Beijing 100094, China;
2. Technology and Engineering Center for Space Utilization, China Academy of Sciences, Beijing 100094, China;
3. Graduate University of China Academy of Sciences, Beijing 100049, China)

Abstract: With the rapid development of space technology, the mission operations system becomes increasingly large, complex, management costs are also increasing. In view of this situation, CCSDS proposed the mission operations system reference model for service-oriented architecture, MO service reference model. The MO reference model is intended to make mission operations system has become the system assembled by a series of small, standardized components. Design of Message Abstraction Layer (MAL) is core of the MO reference model. Take needs of payload operation system as the starting point in this article, while based on the MO MAL reference, combined with object-oriented methodology, design patterns and so on, MAL is designed.

Key words: mission operations system; MAL; SOA; design patterns

0 引 言

任务运控系统是航天地面系统中的重要组成部分。随着我国的空间技术发展,以及未来空间站建设的需求,任务运控系统将变得更加复杂、庞大。由于航天地面系统的特殊性,任务运控系统需要边搭建边应用,其对系统的扩展性要求很高。

任务运营服务(Mission Operations Services)是由空间数据系统咨询委员会(CCSDS)的飞行器监控工

作组(SM&C)提出的一个基于 SOA 的任务运营管理体系架构概念。此架构不仅使运控系统拥有良好的扩展性,并且可以增强各个组件的重用性,因此可以减少系统的开发以及维护成本^[1~4]。

文中基于 MO 服务概念在任务运控系统中的应用需求,结合面向对象方法学、面向服务的体系架构(SOA)设计原则^[1~3]以及设计模式^[5]等思想理念,给出任务运控系统的消息抽象层的设计与实现。

收稿日期:2012-10-26

修回日期:2013-01-30

网络出版时间:2013-04-22

基金项目:中国载人航天项目(Y2140104SN)

作者简介:刘 宇(1987-),男(回),北京人,硕士研究生,研究方向为航天地面支持系统;刘玉荣,硕士,副研究员,研究方向为航天地面系统技术。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20130422.1722.039.html>

1 MO 服务参考模型

CCSDS 在标准 520.0-G-3 中对 MO 服务参考模型进行了描述^[6]。MO 参考模型定义了两个重要的方面:其一是两个实体间的交互模型,其二是具有公共功能的公共服务框架。任务运营服务框架如图 1 所示,分为四层,自上至下分别是应用层、MO 服务层(MO Services Layer)、消息抽象层(Message Abstraction Layer, MAL)、传输层。

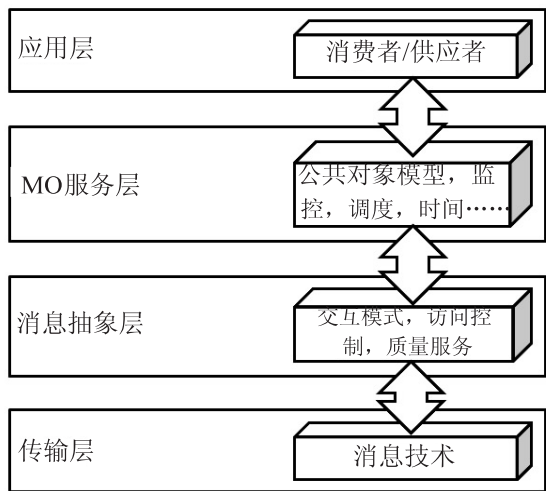


图 1 任务运营服务框架

应用层是由应用程序组成,这些应用程序是针对不同任务或不同载荷而设计的。而应用程序的开发则依赖于服务层所提供的服务,通过调用、组合不同的服务,可以方便地开发出各种应用程序。消息抽象层为服务层提供了各种通信交互模式、访问控制以及服务质量等服务,使服务层可以专注于业务关系,而不需要考虑客户端与服务端的交互方式。传输层则为消息抽象层提供消息编码、解码以及传输方式等服务。

以航天地面系统中卫星数据监视显示功能为例,数据监显软件处于应用层,是服务的消费者,其通过调用服务层提供的服务接口获取其关注的信息;卫星数据处理程序也处于应用层,是服务的供应者,其通过服务层提供的接口向外发布其处理结果。服务端和客户端之间需要通过消息抽象层进行消息交互。消息在发送端发送出去之前需要在传输层进行编码,在接收端接收后需要进行解码,并将消息内容返回给消息抽象层进行处理。具体的传输协议也由传输层定义,例如,可以使用 Tcp、Udp 等方式进行底层的消息传输。

2 MAL 交互模式设计

消息抽象层是为任务运营服务提供基础服务模式的一个框架,在参考文献[6]中有对其的定义。其服务消费者和供应者之间提供了一个标准消息传递

层,从而确保可以通过服务接口标准化绑定 MO 服务层中不同实现的互操作,并为 MO 服务层消费者和供应者两边提供底层通信协议栈,即通信交互模式^[7]。这些交互模式包括发送(Send)、提交(Submit)、请求(Request)、调用(Invoke)、前进(Progress)以及发布-订阅(Publish-Subscribe)交互模式^[8]。

不同的交互模式应对不同的运控场景,例如:在数据归档服务中,就可以采用提交交互模式,客户端程序将需要归档的信息提交给归档服务程序,归档服务程序在收到归档消息后,根据归档结果向客户端返回归档确认消息;在卫星数据监显中,就可以采用发布-订阅模式,由数据处理服务向外发布其可以提供的信息,监显终端根据自己的需要订阅指定的数据信息。

2.1 发送交互模式

发送模式是最基础的交互模式,其他交互模式可在此模式基础上进行扩展。其过程为从消费者端(Consumer)发送一个简单的消息到供应者端(Provider),供应者不会向消费者发送确认信息(ACK)。

2.2 提交交互模式

提交模式扩展自发送交互模式,其具体形式是在消费者向供应者发送消息后,供应者会向消费者发送确认信息。

如果在操作处理中发生错误,则返回的确认信息将会被错误消息取代。对于一次提交操作,确认消息和错误消息不会同时返回。确认消息只有消息头,没有消息体,尽可能使得提交操作简单。错误消息与确认消息格式一致,但是在错误消息的消息头中 IsErrorMessage 字段为 true。

2.3 请求交互模式

请求交互模式扩展自提交交互模式,其将提交交互模式中的简单确认消息替换为带有数据信息的应答消息。

如果在操作处理中发生错误,则返回的响应信息将会被错误消息取代。对于一次请求操作,响应消息和错误消息不会同时返回。错误消息与响应消息格式一致,但是在错误消息的消息头中 IsErrorMessage 字段为 true,错误消息的消息体中存放错误信息。

2.4 调用交互模式

调用交互模式扩展自请求交互模式,其将为请求交互模式中的初始化消息增加强制的确认信息。这种方式使得操作在处理请求前可以得到请求接收确认,其交互模式如图 2(a)。

调用交互模式可以通过两种不同的方式报告错误。第一种方式,供应者将返回一个错误消息替代确认消息。第二种方式,供应者将在发送完确认消息后返回一个错误消息来取代响应消息。在错误消息发送

后,此模式状态将终结,不再进行余下的交互操作。

2.5 前进交互模式

前进交互模式扩展自调用交互模式,其增加了一系列对请求消息的更新。此模式使得被调用的操作在开始处理收到的请求前确认收到此请求消息,而且也可以显示出此操作的执行过程,其交互模式如图 2 (b)。

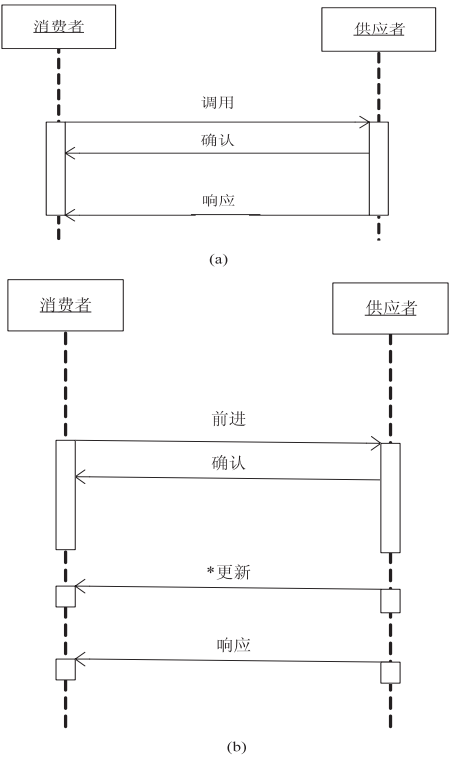


图 2 调用交互模式和前进交互模式

前进交互模式可以通过三种不同的方式报告错误。第一种方式,确认信息可能会被一个错误消息所取代。第二种方式,当初始的确认信息发送后,操作体的处理过程中可能发生错误,从而更新消息将会被错误消息替代。第三种方式,在更新消息后发生错误,从而最后的响应消息将被错误消息取代。在错误消息发送后,此模式状态将终结,不再进行余下的交互操作。

2.6 发布-订阅交互模式

发布-订阅交互模式提供了一种由消费者可以对自己感兴趣的话题进行注册,并可以在不了解供应者信息的情况下从供应者那里接收自己订阅内容的更新消息。

此模式可以描述为:消费者通过发送订阅列表来注册自己感兴趣的内容;供应者也注册他们提供的实体 (entity) 列表;供应者发布更新消息;更新消息随后被发送到注册此内容的消费者;当消费者不再需要更新消息的时候,消费者可以通过注销 (deregister) 操作来取消订阅。

发布-订阅模式可以有两种部署方式:

第一种部署方式是在消费者与供应者之间存在一个中间消息共享的中间代理 (Broker), 如图 3 (a)。这种方式使得消费者与供应者之间解耦,中间代理负责接收消费者的订阅消息以及供应者的更新消息,并且负责分发更新消息给需要更新消息的消费者。

第二种部署方式是用一个对供应者来说是私有的中间代理。这种方式更类似于观察者模式。在这种部署方式中,由于消费者和供应者是一对一的关系,所以这种消息代理的功能不再是必需的了。消息代理的功能完全可以归属到供应者应用程序中。供应者给自己发布更新并通知消费者,如图 3 (b)。

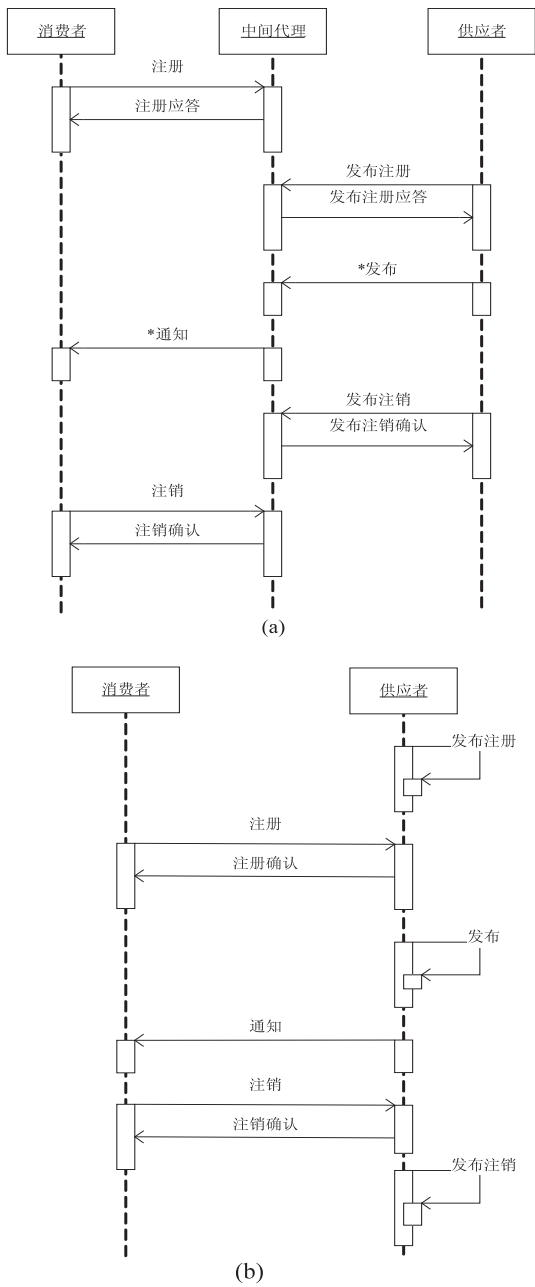


图 3 发布-订阅交互模式

对于发布-订阅模式中的消费者有两种报告错误的方式。第一种方式,注册确认信息可能会被一个错

误消息所取代。第二种方式,一个通知消息可能被中间代理的错误消息取代。一个在错误消息发送后,此模式状态将终结,不再进行余下的交互操作。

对于发布-订阅模式中的供应者有两种报告错误的方式。第一种方式,发布注册确认信息可能会被一个错误消息所取代。当发布注册错误消息发送后,模式中有关发布的消息将不再被传输。第二种方式,当中间代理检查出由供应者发送的发布消息有错误时,中间代理将向供应者发送一个发布错误消息。

3 MAL 接口设计

MAL 接口由五部分组成^[9,10]:

(1) 每个 MAL 客户端使用的通用 API。对于消费者端或供应者端都需要按照给定的交互模式进行交互,这个交互模式的接口就是每个客户端的通用 API。

(2) 数据结构和相关接口。在 MAL 接口中各个类或接口之间的通信需要依赖一些特定的数据结构或接口对象。

(3) 服务消费者 API。MO 服务层的客户端需要通过调用服务消费者 API 才可以实现与服务端的通信。

(4) 服务供应者 API。MO 服务层的服务端需要通过调用服务供应者 API 才可以实现为客户端提供服务。

(5) 中间代理 API。主要用于发布-订阅模式,可以实现消费者和供应者的各种交互操作。

本节将主要介绍服务消费者 API、服务供应者 API 以及数据结构和相关接口设计。

3.1 相关接口及数据结构

1) MALFactory 类。

MALFactory 类能够为 MAL 客户端实例化和配置 MAL 对象。此外, MALFactory 类两个静态资源:

a) Element 类资源。其通过 Element 的短名字(shortname)来映射实际类。

b) MALArea 资源,其通过区域名来映射 MALArea 的描述信息。

MALFactory 类提供一个静态方法,此静态方法返回 MALFactory 的一个实例。这个方法可根据配置信息,运用反射机制将指定的 MALFactory 实现类实例化。MALFactory 类还将提供一个抽象的公共方法来实例化 MAL 对象,此方法需要传递参数为 MAL 实现所需要的资源,例如,主机名或端口号等。

2) MAL 接口。

MAL 接口为客户端提供可用的通信功能上下文。MAL 的实现类为 MALConsumerManager 对象和 MALProviderManager 对象提供创建方法。

3) MessageHeader。

MessageHeader 用于描述消息头信息,其中包括^[11]:

- a) 消息的来源,即消息发送端的 URI;
- b) 消息的去处,即消息接收端的 URI;
- c) 时间戳,即消息的发生时间;
- d) 服务质量等级;
- e) 优先级;
- f) 消息所属区域;
- g) 会话类型;
- h) 交互模式类型;
- i) 交互模式中所处的交互阶段;
- j) 消息所属的服务;
- k) 消息所对应的操作;
- l) 是否是错误消息标志。

3.2 Consumer 接口

1) MALConsumerManager 接口。

MALConsumerManager 对象将 MAL 消费者初始化交互模式和接收服务供应者响应消息所需的资源封装起来。MALConsumerManager 是 MALConsumer 对象的创建工厂。在创建 MALConsumer 时需要提供以下参数:

- a) 服务供应者的 URI;
- b) 中间代理的 URI,用于发布-订阅模式;
- c) 服务描述;
- d) MALConsumer 与 MALProvider 交互过程中的标识;
- e) 服务供应者所属区域;
- f) 服务供应者所属网络;
- g) 服务会话类型,如实时、回放或者模拟;
- h) 会话名;
- i) 消费者与供应者所有交互所需的服务质量(QoS)等级;
- j) 服务质量等级所需资源,例如 timeToLive,即消息存活时间。

2) MALConsumer 接口。

MALConsumer 对象为客户端提供初始化交互模式(其中包括同步和异步两种方式)的通信上下文。MALConsumer 为每个交互模式提供一个同步和异步方法。例如,发布-订阅的注册的同步方法:

```
public void register(MALPubSubOperation op, Subscription subscription, MALInteractionListener listener)
```

其中 MALSubmitOperation、MALRequestOperation、MALInvokeOperation、MALProgressOperation、MALPubSubOperation 继承自基类 MALOperation,用于描述服务中各种交互模式下的各种操作信息。

Subscription 是消费者在发布-订阅模式中要订阅的内容的信息。

3)MALInteractionListener 接口。

MALInteractionListener 接口由 MAL 客户端实现,用于异步接收供应者发送的消息。其定义了对确认消息、响应消息、更新消息、通知消息以及错误消息的接收方法。每个方法都需要提供参数操作方法信息,即 MALOperation;消息头内容 MessageHeader;消息体内容,消息体根据不同的交互模式会有所不同。例如,错误消息接收方法定义如下:

```
public void errorReceived ( MALOperation op, MessageHeadermsgHeader, StandardError error)
```

其中 errorReceived 用于接收错误消息,StandardError 将错误信息封装在其内部,并将错误信息报告给 MAL 客户端。

3.3 Provider 接口

1) MALProviderManager 接口。

MALProviderManager 对象将 MAL 供应者的控制交互模式和给消费者发送响应所需的资源封装起来。MALProviderManager 对象为 MALProvider 分配 URI。MALProviderManager 是 MALProvider 对象的创建工厂。在创建 MALProvider 时需要提供以下参数:

- a) 服务供应者的名字,此名字必须是唯一给定的;
- b) 协议名,用于绑定服务供应者;
- c) 服务描述;
- d) MALProvider 的标识;
- e) 交互模式控制对象;
- f) 供应者所需的服务质量;
- g) 中间代理的相关信息。

2)MALProvider 接口。

MALProvider 对象描述一个给定 URI 的服务供应者的执行上下文。MALProvider 将提供方法返回自己的 URI 和中间代理的信息等。由于大部分交互的发起都是由消费者发起,所以控制进程采用“推”的方式将交互申请推给服务供应者。

3)MALInteractionHandler 接口。

实现 MALInteractionHandler 接口的对象主要负责初始化 MALProvider 的交互控制。除发布-订阅模式,其应对每一个交互模式定义一个控制方法。例如,请求交互模式:

```
public void handleRequest ( IMALRequest interaction,Element body)
```

4)MALInteraction 接口。

MALInteraction 为所有交互控制对象的父接口。其拥有对应操作的相关信息和初始化交互模式的

MessageHeader 的引用。

其子接口包括 MALSubmit、MALRequest、MALIn-voke 以及 MALProgress,继承关系如图 4 所示。

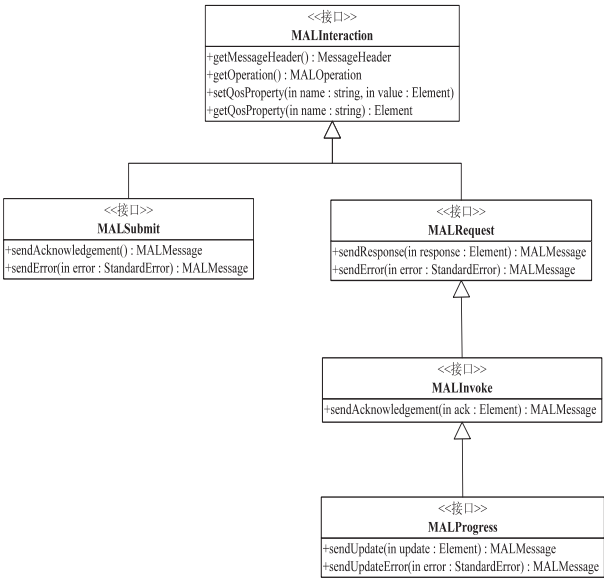


图 4 MALInteraction 继承关系

4 在系统中的应用

在数据监显程序中,数据监显软件向数据处理软件订阅不同分系统或不同载荷的数据。而数据处理软件在收到数据后,对各个数据进行处理,并将各个数据的处理结果分发给订阅它们的数据监显程序,数据监显程序将处理结果显示给用户。数据服务框架如图 5 所示。

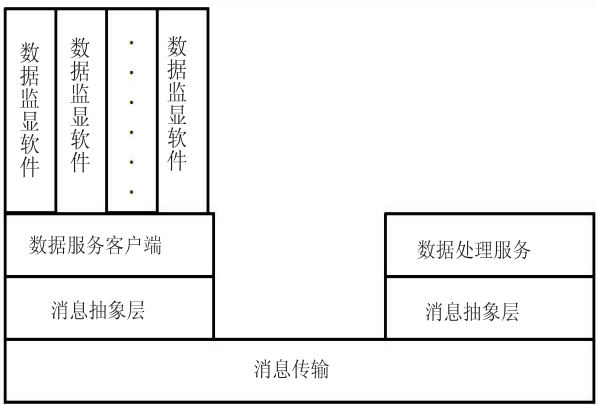


图 5 数据服务框架

5 结束语

文中主要介绍对 MAL 层的接口设计,通过完善 MAL 层接口,可以方便地开发出服务层的服务程序,从而达到扩展任务运营系统功能的目的。例如,增加载荷或者飞行器的情况下,可以在不修改运营管理系统的前提下,将新的服务程序加入到系统中去,应用程

率能达到 95.2%,但是只取其中一个特征或是某几个特征组合时,识别率明显降低。

表 1 所有实验数据准确率

数据来源	特征组合	准确率
由 2.2 得到的所有角度和距离下的实验数据	$x_1、x_2、x_3、x_4、x_5、x_6$	95.21%
	$x_1、x_3、x_4$	93.13%
	$x_2、x_3、x_4$	94.89%
	$x_1、x_5、x_6$	92.97%
	$x_1、x_2、x_3、x_4$	93.93%
	$x_5、x_6$	92.97%
	x_1	92.01%
	$x_3、x_4$	91.65%

(3)在线应用。

将建立的决策树模型,应用于基于 STM32 微处理器的多红外火焰探测系统并编写相关的决策树模型程序。经过 50 次实验验证,其中有 46 次实验得到的正确率能达到 95%,并且响应速度在 2s 以内。而传统的火灾探测的识别算法火灾识别准确率在 85% 左右,响应速度约为 10s。因此,该决策树模型在多红外火焰系统中的应用具有较高的实用价值。

4 结束语

文中研究的决策树算法在多红外火灾识别中的应用,将多个探测器探测到的火焰信息,进行加窗预处理,找到六个作为火灾判据的特征并建立决策树模型,可以很方便的进行分类决策,得到最优的决策值。此外,剪枝技术在决策树中的应用,检验了决策树中规则的稳定性,去掉了决策树中不稳定的规则,使得得到的决策树减少了由于过度拟合而产生的偶然规则,提高了决策树的泛化能力。通过 N 倍交叉验证,将大量样

+++++
(上接第 13 页)

序可以方便地使用服务程序提供的 API 进行开发,从而降低应用程序的开发和系统维护成本。

参考文献:

[1] 林怀恭,聂瑞华,罗辉琼,等. 基于 SOA 架构的服务集成技术的研究[J]. 计算机技术与发展,2009,19(7):141-144.

[2] 杜攀,徐进. SOA 体系下细粒度组件服务整合的探讨[J]. 计算机应用,2006,26(3):699-702.

[3] 卢致杰,覃正,韩景倜,等. SOA 体系设计方法研究[J]. 工业工程,2004,7(6):14-19.

[4] 张晖,刘亚璠,黄翰榕,等. 遥感卫星地面系统面向服务流程管理研究[J]. 无线电工程,2009,39(11):4-6.

[5] Gamma E,Helm R,Johnson R,et al. 设计模式:可复用面向对象软件的基础[M]. 李英军,译. 北京:机械工业出版社,

本集进行训练和测试,得到了在不同情况下系统的识别率。较之前的探测识别率有了较大的提高,通过反复实验验证了其可行性,识别率能达到 95.2%,具有很高的应用价值。

参考文献:

[1] 张红兰,李扬. 基于多传感器的智能火灾报警器的设计[J]. 仪器仪表学报,2009,28(4):48-50.

[2] Xu Xiaochun, Sahni S. Approximation Algorithms for Sensor Deployment[J]. IEEE Transactions on Computers,2007,56(12):1681-1695.

[3] 习树峰,彭勇,梁国华,等. 基于决策树方法的水库跨流域引水调度规则研究[J]. 大连理工大学学报,2012,52(1):74-78.

[4] 李任良,李义杰. 基于多策略的决策树剪枝算法及其应用[J]. 计算机仿真,2010,27(11):78-81.

[5] 许惠君,李彩林,刘晓安. 数据挖掘技术在水库调度中的研究与应用[J]. 计算机与数字工程,2006,34(9):61-63.

[6] 曲开社,成文丽,王俊红. ID3 算法的一种改进算法[J]. 计算机工程与应用,2003,39(25):104-107.

[7] 纪希禹,韩秋明,李微,等. 数据挖掘技术应用实例[M]. 北京:机械工业出版社,2009.

[8] Wang B,Kee C C,Srinivasan V,et al. Information Coverage in Randomly Deployed Wireless Sensor Networks[J]. IEEE Transactions on Wireless Communications,2007,6(8):2994-3004.

[9] 孙爱东,朱梅阶,涂淑琴. 基于属性值的 ID3 算法改进[J]. 计算机工程与设计,2008,29(12):3011-3012.

[10] 王会青,陈俊杰,侯晓晶,等. 决策树分类的属性选择方法的研究[J]. 大连理工大学学报,2011,42(4):346-348.

[11] Scherzinger H. Requirements for Fire Detectors - Industrial Application[J]. Springer Series on Chemical Sensors and Biosensors,2011,10(5):1781-1789.

+++++
2007.

[6] Mission Operations Services Concept[S]. CCSDS 520.0-G-3 Green Book,2010.

[7] Mission Operations Reference Model[S]. CCSDS 520.1-M-1 Magenta Book,2010.

[8] Mission Operations Message Abstraction Layer[S]. CCSDS 521.0-B-1 Blue Book,2010.

[9] MO JAVA MAL API[S]. CCSDS 000.0-R-0 Red Book,2010.

[10] Mission Operations Message Abstraction Layer-Java API[S]. CCSDS 521.2-R-1 Red Book,2011.

[11] Mission Operation Common Object Model[S]. CCSDS 521.1-R-3,2012.

基于CCSDS M0服务参考模型消息抽象层设计

作者:	<u>刘宇, 刘玉荣, LIU Yu, LIU Yu-rong</u>
作者单位:	<u>刘宇,LIU Yu(中国科学院 光电研究院, 北京 100094; 中国科学院 空间应用工程与技术中心, 北京 100094; 中国科学院 研究生院, 北京 100049), 刘玉荣,LIU Yu-rong(中国科学院 空间应用工程与技术中心, 北京, 100094)</u>
刊名:	<u>计算机技术与发展</u>
	<div>ISTIC</div>
英文刊名:	<u>Computer Technology and Development</u>
年, 卷(期):	<u>2013(8)</u>

本文链接: http://d.g.wanfangdata.com.cn/Periodical_wjtz201308003.aspx