

# 面向 RFID 系统的复杂事件处理机制与方法

## Mechanism and Method for Handling Complex Event in RFID System

康普科纬软件服务(上海)有限公司 张 宇

[摘要] 总结了供应链RFID系统中复杂事件处理的应用,在目前研究成果的基础上对复杂事件相关概念及定义进行了完善,对复杂事件表达式的运算性质进行了分析,在此基础上给出了复杂事件检测算法,并对其性能进行了分析。

关键词: 供应链 RFID 复杂事件处理

[ABSTRACT] Summarize the complex event in the RFID system of supply chain management, and improve the relating concept and definition of complex event based on the current research. Analyse the property of the complex event expression. At last, complex detecting algorithm is given and the performance is tested.

**Keywords: Supply chain RFID Complex event processing**

在现代的企业管理中,无线射频识别(Radio Frequency Identification, RFID)中间件平台在供应链管理系统(Supply Chain Management, SCM)中得到了广泛的应用。将RFID标签粘贴在托盘、包装箱或元器件上,可以将信息传递给一定范围内的读卡器,使配送、质检、入库、盘点和出库得以实现自动化管理,同时对货物配送路径的管理,货物的追踪溯源也提供了很大的便利<sup>[1]</sup>。

在日常业务中,标签卡通过读卡器产生大量的原始数据,而原始数据对管理者并没有太大的价值,企业管理者真正需要的是原始数据背后蕴含的商业逻辑以及可以帮助进行决策的高级信息。基于供应链RFID平台的复杂事件处理技术的引入,使得系统可以自动地从原始事件流中提取出有价值的商业信息并触发相关的操作,操作者可以根据自己的权限和兴趣订阅相关的信息<sup>[2-4]</sup>。

目前有很多研究者把目光投向了复杂事件处理的研究,并取得了很好的成果<sup>[5-8]</sup>,其中文献[5]对复杂事件进行了定义,提出了SASE复杂事件检测算法,文献[6]对复杂事件的描述语言和检测算法进行了分析。文献[7-9]对复杂事件检测算法进行了改进,目前存在的主要

问题是:①对复杂事件描述语言的性质没有进行深入分析;②复杂事件检测对算法的实时性有较高的要求,目前的算法需要大量的回溯,性能有待提高。

### 1 复杂事件描述语言 CDL

#### 1.1 CDL 描述语言句型定义

复杂事件描述语言(Complex-Event Description Language, CDL)可以准确地对事件流进行描述、匹配、过滤和转换,可以匹配原始事件流中基于时间和条件因素形成的各种复杂事件,并且定义匹配成功后将触发的动作和事件,其语法结构为:

EVENT < 事件表达式 exp>

WHERE < 条件表达式 q>

[AT, WITHIN, BETWEEN] < 时间表达式 t>

TODO < 动作表达式 a>

其语义为:当事件表达式 exp 在时间表达式 t 规定的时间范围内满足条件表达式 q 描述的条件约束时,触发动作表达式 a 描述的内容。

例 1 给出了复杂事件描述语言 CDL 的具体实例。

例 1: EVENT (A a) && (B b) && (C c)

WHERE a.id = b.id  $\wedge$  a.id = c.id

WITHIN 12hours

TODO ALERT

上述例子描述的是,在 12h 内,如果 3 种 id 相同的事件(A, B, C)均发生时则触发警报。

下面分别对 EVENT 事件表达式、WHERE 条件表达式、[AT, WITHIN, BETWEEN] 时间表达式以及 TODO 动作表达式进行详细描述。

#### 1.1.1 事件表达式及其性质

文献[5]中定义了事件类型和事件实例,本课题在此基础上进行了补充,并对其性质进行了深入分析。以下是具体的定义和详细分析。

用大写字母表示一个事件的类型,如 E=(ID, ATTRS),其中 ID 表示事件类型的编号,ATTRS={attr1, attr2, ..., attrn}, $n \geq 1$  是属性集合表示复杂事件的各

个相关属性值。

用小写字母表示特定事件类型的一个事件实例, 如  $e = (ID, TYPEID, TIME_{begin}, TIME_{end})$ , 其中 ID 表示事件实例编号, TYPEID 表示事件类型编号,  $TIME_{begin}$  和  $TIME_{end}$  分别表示复杂事件起始和结束的时间, 对于简单事件来说两者值相等, 每一个事件实例属于且只属于一个事件类型。

下面定义事件表达式并分析其性质。

定义 1. 事件表达式: 利用事件类型、事件实例和各种连接符表达业务系统中发生事件的表达式。

性质 1. 如果  $Exp_1$  是事件表达式, 则  $(Exp_1)$  是事件表达式。

性质 2. 如果  $Exp_1, Exp_2$  都是事件表达式, 则  $(Exp_1 \&\& Exp_2)$  是事件表达式。

性质 3. 如果  $Exp_1, Exp_2$  都是事件表达式, 则  $(Exp_1 || Exp_2)$  是事件表达式。

性质 4. 如果  $Exp_1, Exp_2, \dots, Exp_n$  都是事件表达式, 则  $ORDER (Exp_1, Exp_2, \dots, Exp_n, n \geq 1)$  是事件表达式。

其中连接符 !, &&, || 和 ORDER 表达式定义如下:

定义 2. 连接符 && (并且): 当 && 连接的 2 个事件都发生时, 表达式值为真。形式化定义为  $(Exp_1 \&\& Exp_2) \equiv Exp_1 \wedge Exp_2$ 。

定义 3. 连接符 || (或者): 当 || 连接的 2 个事件之一发生时, 表达式值即为真。形式化定义为:  $(Exp_1 || Exp_2) \equiv Exp_1 \vee Exp_2$ 。

定义 4. 连接符 ! (否定): 连接符起到否定的作用, 加在事件类型前表示该事件类型不发生。带有 ! 连接符的事件称为否定事件。形式化定义为:  $!(Exp_1) \equiv \neg Exp_1$ 。

性质 5.  $!(Exp_1 \&\& Exp_2)$  等价于  $!Exp_1 || !Exp_2$ 。

性质 6.  $!(Exp_1 || Exp_2)$  等价于  $!Exp_1 \&\&!Exp_2$ 。

供应链中的事件之间是有时间上的先后顺序的, 按照特定的时间顺序发生, 用次序表达式描述。

定义 5. 肯定 ORDER 表达式和否定 ORDER 表达式统称为 ORDER 表达式。

当 ORDER 表达式不包含否定事件时, 称其为肯定 ORDER 表达式, 表示多个事件按顺序发生。可用形如  $ORDER (Exp_1, Exp_2, \dots, Exp_n, n \geq 1)$  的表达式来描述。

当 ORDER 表达式包含否定事件时, 称其为否定 ORDER 表达式, 当表示  $Exp_2$  不在  $Exp_1$  和  $Exp_3$  之间发生时可以用形如  $ORDER (Exp_1, !(Exp_2), Exp_3)$  的表达式来描述。

性质 7.  $ORDER (Exp_1, Exp_2, Exp_3) \cap ORDER (Exp_1,$

$!Exp_2, Exp_3) = \emptyset$ 。

性质 8.  $ORDER (Exp_1, Exp_2, Exp_3) \cup ORDER (Exp_1, !Exp_2, Exp_3) = ORDER (Exp_1, Exp_3)$ 。

可以根据性质 7 和 8 推导出一个重要结论: 当需要得到否定 ORDER 表达式  $ORDER (Exp_1, !Exp_2, Exp_3)$  的事件集合时, 只需要得出 2 个肯定 ORDER 表达式的集合, 即用  $ORDER (Exp_1, Exp_3)$  的事件集合减去  $ORDER (Exp_1, Exp_2, Exp_3)$  的事件集合就可以得到  $ORDER (Exp_1, !Exp_2, Exp_3)$  的事件集合, 从而可以简化 ORDER 句型的检测算法。

定义 6. 直接前驱事件和直接后继事件在 ORDER 表达式中是有次序关系的, 对相邻的 2 个事件, 将前一个事件称为后一个事件的直接前驱事件, 将后一个事件称为前一个事件的直接后继事件。

定义 7. 最简 ORDER 表达式: 当 ORDER 表达式不含有 && 并且不含有 || 时, 该表达式称为最简 ORDER 表达式。而  $ORDER (Exp_1, (Exp_2 \&\& Exp_3), Exp_4)$  以及  $ORDER (Exp_1, (Exp_2 || Exp_3), Exp_4)$  不属于最简 ORDER 表达式。

性质 9.  $ORDER (Exp_1, (Exp_2 \&\& Exp_3), Exp_4)$  等价于  $ORDER (Exp_1, Exp_2, Exp_3, Exp_4) || ORDER (Exp_1, Exp_3, Exp_2, Exp_4)$ 。

性质 10.  $ORDER (Exp_1, (Exp_2 || Exp_3), Exp_4)$  等价于  $ORDER (Exp_1, Exp_2, Exp_4) || ORDER (Exp_1, Exp_3, Exp_4)$ 。

性质 9 和 10 可以将复杂的 ORDER 表达式化为多个简单的 ORDER 表达式, 以便简化 ORDER 句型检测算法。

### 1.1.2 条件表达式的定义

事件表达式中出现的每一个事件实例都是有属性的, 所谓条件表达式就是对事件实例的属性间关系加以测试, 对事件表达式中的事件实例加以逻辑限制的一种表达式。

### 1.1.3 时间表达式的定义

时间表达式用于限定事件发生的时间范围。

当表示在一个时间点发生的事件时, 用关键字 AT, 后接一个表示某一时刻的参数; 当表示在一个时间段内发生的事件时, 用关键字 WITHIN, 后接一个表示时间段大小的参数; 当表示 2 个时间点之间发生的事件时, 用关键字 BETWEEN, 后面接开始时间和结束时间 2 个参数。

### 1.1.4 动作表达式的定义

在原始事件流上, 当事件表达式、条件表达式和时间表达式的限定全部满足时, 即触发动作表达式规定的内容。动作表达式由业务系统定义, 如例 1 中的 ALERT

表示报警,业务系统中的其他动作也可以定义相应的表达式。

## 1.2 CDL 描述语言应用举例

CDL 描述语言可以应用在 ERP 和 SCM 的 RFID 系统中,下面以供应链 RFID 系统为例,介绍 CDL 语言的具体应用。

在供应链 RFID 系统中,厂房和仓库的入口门禁读写器、出口门禁读写器、货架上安装的读写器以及一定数量的手持读写器构成了一个 RFID 读写器网络。

例 2: EVENT ORDER (ENTRANCE-READING x, ! (CHECK-READING y), EXIT-READING z)

WHERE x.id = y.id  $\wedge$  x.id = z.id

WITHIN 12hours

TODO report\_to\_manager

例 2 中的 ENTRANCE-READING 表示仓库入口处读卡器产生的入库事件, CHECK-READING 表示质检事件, EXIT-READING 表示出库事件, x, y, z 分别是 3 个事件实例。描述的意义是检测符合以下条件的产品:在 12h 内完成入库操作,没有经过质检就通过了仓库出口。如检测到结果,说明有产品发生了漏检,即向管理员发送报告。

通过以上例子说明 CDL 语言可以准确地描述供应链 RFID 系统的日常业务,以及在规定的时间内满足条件时触发的动作。CDL 也可以较好地描述其他 RFID 系统的日常业务。

## 2 复杂事件检测算法

供应链系统的日常业务中,大量 RFID 原始事件实例不断地涌入中间件平台,对复杂事件处理的速度、效率及实时性提出了很高的要求,虽然文献 [5] 提出了 SASE 复杂事件检测算法,但以下 2 点尚有待改进:

① 在原始数据流上进行大量回溯,当数据流大时,无法保证处理速度,需要优化以提高效率。

② 实时处理要求高效率,算法需要进一步优化以提高处理速度。

本课题给出一个复杂事件检测算法 CD (Complex-Event Detecting Algorithm),可以适应各种复合事件表达式,为复杂事件提供了高效率的检测,从而提高了算法的效率。

### 2.1 存储数据流的数据结构

设系统中某一时刻的事件流 S 为

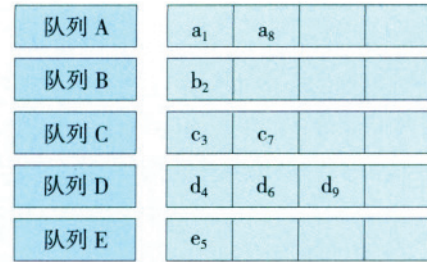
$$\frac{\text{事件 } a \ b \ c \ d \ e \ d \ c \ a \ d \ \dots}{\text{时间 } 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ \dots} \circ$$


图 1 事件流 S 经过预处理后产生的事件队列

Fig.1 Event queue of event flow

复杂事件检测算法对接收到的数据流要进行预处理,数据结构和预处理过程见图 1。

系统为每一种事件类型创建 1 个队列,称为事件队列。每当接收到 1 个 RFID 事件实例,如果不存在这种事件类型的事件队列,则新增相应的事件队列;如果存在,则将事件实例插入到相应事件队列的尾部。这样,RFID 中间件平台产生的每个事件都归结到事件相应的队列中,而每个队列都是按照事件发生的时间顺序排列的,这样可以为检索及进一步处理打下基础。

### 2.2 ORDER 表达式句型的复杂事件检测算法

#### 2.2.1 肯定 ORDER 句型检测算法

下面举例说明肯定 ORDER 句型的检测算法。

例 3: EVENT (ORDER(A a, B b, C c, D d)

WHERE a.id = b.id  $\wedge$  b.id = c.id  $\wedge$  c.id = d.id

WITHIN 4 hours

TODO ReportToManager

A, B, C, D 代表 4 个不同的事件类型,这个复杂事件查询表达式的意义是,在 4h 内,如果一个产品连续发生了 A, B, C, D 4 种类型的事件,则将此事件报告给管理员。

首先,对队列中的每一个事件,如果有直接后继事件,则将它与此后发生的直接后继事件链接起来,形成后继事件链表。形成的结果如图 2 所示。

链表中每一条从起点到终点的路径都显示了事件

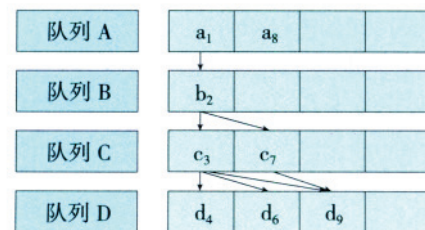


图 2 建立后继事件链表

Fig.2 Chain of subsequence event

流 S 中符合过滤规则的事件。本例中得出的结果是  $(a_1, b_2, c_3, d_4)$ ,  $(a_1, b_2, c_3, d_8)$ ,  $(a_1, b_2, c_3, d_9)$  和  $(a_1, b_2, c_7, d_9)$ 。

最后,对得到的几组事件序列进行条件过滤和时间窗口过滤,例 3 中规定的窗口大小是 4h,只有第一级  $(a_1, b_2, c_3, d_4)$  符合过滤规则,系统触发 TODO 事件,并返回  $(a_1, b_2, c_3, d_4)$ 。

### 2.2.2 否定 ORDER 句型检测算法

当 ORDER 表达式中包含否定事件时,不能使用以上算法来检测,但可以根据上文提到的性质 7 和 8,将它转化为 2 个肯定 ORDER 表达式的差的形式,然后分别算出这 2 个表达式的结果,从而算出原式的结果。

下面举例说明否定 ORDER 句型的检测算法。

```
例 4:EVENT (ORDER (A a, B b, !E e), C c, D d)
WHERE a.id =b.id ^ b.id=c.id ^ c.id=d.id
WITHIN 15hours
TODO ALERT
```

设系统中某一时刻的事件流 S 为:

事件	a	b	c	d	e	d	c	a	d	...
时间	1	2	3	4	5	6	7	8	9	...

在例 4 中,ORDER 句型中包含否定事件,根据性质 7 和 8,只需用上节提出的肯定 ORDER 句型检测算法就可得到以下 2 个子事件流  $Q_1, Q_2$ ,然后将它们 2 个返回的子事件流集合相减,就可以得出正确结果。

```
Q1:EVENT (ORDER (A a, B b, C c, D d))
WHERE a.id =b.id ^ b.id=c.id ^ c.id=d.id
WITHIN 15hours
TODO ALERT

Q2:EVENT (ORDER (A a, B b, E e, C c, D d))
WHERE a.id =b.id ^ b.id=c.id ^ c.id=d.id
WITHIN 15hours
TODO ALERT
```

本例中,  $Q_1$  的结果集为  $(a_1, b_2, c_3, d_4)$ ,  $(a_1, b_2, c_3, d_8)$ ,  $(a_1, b_2, c_3, d_9)$  和  $(a_1, b_2, c_7, d_9)$ ,  $Q_2$  的结果集为  $(a_1, b_2, c_7, d_9)$ ,相减得出结果  $(a_1, b_2, c_3, d_4)$ ,  $(a_1, b_2, c_3, d_8)$  以及  $(a_1, b_2, c_3, d_9)$ ,即为最终结果。

### 2.3 逻辑表达式的复杂事件检测算法

其他句型(如  $(Exp_1 \&\& Exp_2)$ ,  $(Exp_1 || Exp_2)$  句型)的复杂事件检测算法相对来说比较简单,按通常的逻辑判断即可得出结论,这里主要讨论对这 2 种句型检测算法的优化。

对  $(Exp_1 \&\& Exp_2)$  句型,当检测表达式  $Exp_1$  为假时不用继续检测,整个表达式一定为假。对  $(Exp_1 || Exp_2)$  句型,当检测表达式  $Exp_1$  为真时不用继续检测,整个表

达式一定为真。

## 3 性能分析

系统由 VC 开发实现,在 Pentium IV 2.4GHz,1G 内存,Windows XP 环境下运行通过。下文叙述对其性能进行测试,并与通用复杂事件检测引擎 SASE<sup>[5]</sup> 进行对比测试。SASE 是一个开放的事件流处理器,目前得到了广泛的关注,在这里将它作为比较的基准。

定义复合事件表达式:人工编写 20 个复合事件表达式,包括 30 种事件类型,作为复杂事件检测算法的输入。

由图 3 可以看到,随着待处理原始事件数量的增大,SASE 算法处理时间增加较多,而 CD 算法增加较少,这主要是因为 SASE 算法运用过程中引入了大量的回溯,当原始事件数量增大时频繁的回溯严重影响了效率,而 CD 算法则能很好地避免回溯,使实时处理的性能得到了保障。

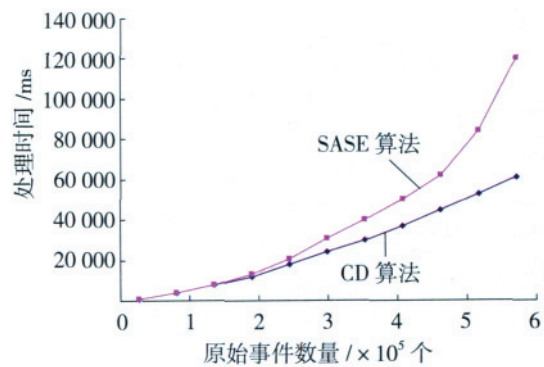


图 3 不同规模原始事件数量情况下处理时间  
Fig.3 Time deal with the number of event

图 4 体现了不同原始事件规模下单位时间内检测到的复杂事件数量,反映了 2 种算法的效率。可以看出,由于回溯,随着原始事件规模增大,SASE 的效率下降得

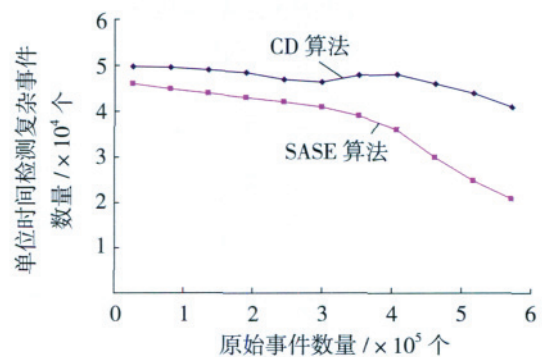


图 4 单位时间检测复杂事件数量  
Fig.4 Number of event in per unit

较快,而 CD 算法受影响较小。

#### 4 结束语

目前,复杂事件处理在 RFID 中间件平台中的应用尚处于理论研究阶段,与实际的结合还有些问题需要解决,其性能测试也没有统一的标准。目前的 RFID 平台普遍只具有记录读卡信息等基本功能,随着 RFID 应用场合的逐渐增多,以及硬件技术的不断成熟,复杂事件处理技术将会得到深入研究,信息系统也将会更加智能化,从而帮助企业管理者和经营者更加高效、准确地进行商业信息的分析与判断。

#### 参 考 文 献

- [1] 游战清,李苏剑等. 无线射频识别技术 (RFID) 理论与应用. 北京: 电子工业出版社.
- [2] Wang F, Liu S, Liu P, et al. Bridging physical and virtual worlds: complex event processing for RFID data streams. The 10th International Conference on Extending Database Technology (EDBT), Munich, Germany, 2006.
- [3] SHARIQ R. Complex event processing beyond active databases: streams and uncertainties. California: Department of EECS of University of California at Berkeley, 2005.
- [4] Wu E, Diao Y, Rizvi S. High-Performance Complex Event Processing over Streams. SIGMOD, Chicago, America, 2006.
- [5] 臧传真,范玉顺. 基于智能物件的制造企业复杂事件处理研究. 计算机集成制造系统, 2003 (6): 107-109.
- [6] Adaikkalavan R, Chakravarthy S. SnoopIB: Interval-based event specification and detection for active databases. ADBIS, Dresden, German, 2003.
- [7] Hwang J H, Balazinska M, Rasin A, et al. High-availability algorithms for distributed stream processing. ICDE, Tokyo, Japan, 2005.
- [8] Li M, Liu M, Ding L, et al. Event Stream Processing with Out-of-Order Data Arrival. DEPSA of ICDCSW Conference, Toronto, Canada, 2007.
- [9] 杨清,徐建良,高德欣. 基于复杂事件处理技术的 RFID 系统数据分析. 微计算机信息, 2006 (26): 179-181.

(责编 恒之)

(上接第 81 页)

#### 4 结束语

本课题利用 BP 神经网络技术,建立了钛合金焊接接头机械性能预测模型,根据实际焊接过程中测得的样本数据,确定了网络的输入输出向量,并对数据样本进行了归一化处理,保证了输入分量的同等重要。

通过对钛合金的钨极氩弧焊焊接接头机械性能的

预测,研究了钛合金成分对焊接接头机械性能的影响规律以及不同元素对各种钛合金机械性能的影响,并且详细分析了铝和钒这 2 种元素对不同机械性能的影响。

#### 参 考 文 献

- [1] 张忠典,李严,何幸平,等. 人工神经网络法估测点焊接头力学性能. 焊接学报, 1997, 18 (1): 1-5.
- [2] 彭金宁,陈炳森,朱平. 焊接工艺参数的神经网络智能设计. 焊接学报, 1998, 19 (1): 19-24.
- [3] Huang Shisheng, Li Di, Song Yonglun. Welding quality control by neural network. 1994 International Conference of International Institute of Welding, Beijing, 1994.
- [4] Lalam S H, Bhadeshia H K D H, Mackay D J C. Estimation of mechanical properties of ferritic steel welds, Part 1: yield and tensile strength. Science and Technology of Welding and Joining, 2000, 5 (1): 135-147.
- [5] Lalam S H, Bhadeshia H K D H, Mackay D J C. Estimation of mechanical properties of ferritic steel welds, Part 2: elongation and charpy toughness. Science and Technology of Welding and Joining, 2000, 5 (2): 149-161.
- [6] Kim S, Son J S, Park C E, et al. A study on the quality improvement of robotic GMA welding process. Robotics and Computer-Integrated Manufacturing, 2003, 19 (6): 567-572.
- [7] Nagesh D S, Datta G L. Prediction of weld bead geometry and penetration in shielded metal-arc welding using artificial neural networks. Journal of Materials Processing Technology, 2002, 123 (2): 303-312.

(责编 恒之)

### IAE 庆祝上海航空公司

#### 接收首架以 V2500 为动力的飞机

IAE 国际航空发动机公司于近日庆祝上海航空公司接收首架 V2500 为动力的飞机。这架 A321 飞机在德国汉堡的空客总装厂交付,意味着上航接收了其在 2008 年范堡罗航展上订购的 10 架飞机中的第一架飞机。

在大中华区所有在役和订购的 A321 飞机上, V2500 目前拥有超过 70% 的市场份额。在 A321 飞机上, V2500 可以提供最低的油耗,最低的噪声值和最少的排放量。

共有大中华区的 16 家航空公司约 370 架在役或订购的飞机配备 V2500 发动机。IAE 也是中国天津空客 A320 总装厂的主要发动机供应商,将为计划总装的头 10 架飞机中的 9 架作动力。

V2500-A5 共有 7 种不同的推力额定值,它们为空客 A319、A320 和 A321 飞机系列和 A319 公务机提供动力装置。全球在役或确认订购的 V2500 发动机超过 5 500 台,拥有 180 多家用户。(本刊记者 凌川)