

文章编号: 1000-8055(2007)02-0261-07

可扩展的推进系统仿真平台: 部分 I - 工程模型构建

曹志松, 于龙江, 朴 英
(清华大学 航天航空学院, 北京 100084)

摘 要: 基于 java 的面向对象建模, 建立了可扩展的推进系统仿真平台(EPSP). 研究了 EPSP 中分级模型建立的机制; 基于组件分解和标准化接口, 提出了 EPSP 中所用的工程模型架构的设计. EPSP 允许部件被其它类似功能的部件替换, 用户可以自己组建推进系统模型, 并进行各学科和各精度的仿真计算. 如何构建 EPSP 的工程模型将被描述, 文中给出了部件工程模型创建的示例, 表明可以利用本平台很方便的架构推进系统模型, 体现了“插件化”的思想, 证明本平台的建模达到了预期的目标.

关 键 词: 航空、航天推进系统; 面向对象建模; 推进系统; 仿真平台

中图分类号: V233.7

文献标识码: A

An extensible propulsion simulation platform Part I - foundations of engineering model

CAO Zhi-song, YU Long-jiang, PIAO Ying

(School of Aerospace, Tsinghua University, Beijing 100084, China)

Abstract: EPSP (extensible propulsion simulation platform) was developed as a Java based object-oriented platform for propulsion system simulation. The architecture represents the hierarchal structure of the engine, its components and substructures using recursive composition. This paper presents the design of a common engineering model framework for use in EPSP. This technique provides an access to build models with increasingly complex elements out of simpler ones. The foundations of the platform's engineering model were described. To test the platform, we can easily construct a component based engineering model, and the “pluggability” of the architecture is shown. The example proves the validity and feasibility of the modeling of the platform.

Key words: aerospace propulsion system; object-oriented modeling; propulsion system; simulation platform

航空推进系统的设计开发耗费巨大, 采用计算机仿真模拟技术参与设计、替代部分物理实验, 可以减少开发时间, 节约成本. 为了对推进系统的机理及性能有更好的认识, 我们需要一个灵活的平台式仿真系统软件, 它能结合先进的多学科和多保真度的分析方法, 可以动态搭建新的推进系

统模型, 并且可以分布式并行计算复杂的任务^[1]. 开展这一仿真平台的研制工作, 对我国推进系统行业打好坚实基础, 实现突破性进展, 进而参与世界竞争有极其重要的意义. 搭建这一平台之初, 框架设计至关重要. 框架设计必须具备先进性、科学性、系统性和完整性. 自 20 世纪 90 年代中后期,

收稿日期: 2006-01-09; 修订日期: 2006-08-06

作者简介: 曹志松(1977-), 男, 江苏靖江人, 清华大学航天航空学院博士生, 主要从事航空发动机系统级仿真研究.

我国的科研机构 and 高等院校在这方面也做了相当工作^[2,3],也取得了很多成果,但在系统级建模的研究上与国外相比还有一些差距.

可扩展的推进系统仿真平台(EPSP)是我们独立开发的基于组件的推进系统仿真平台,它可以用于推进系统性能预测和非设计点性能的分析,也适用于推进系统对诸如飞行环境、控制规律变化、部件设计参数等方面的灵敏性分析(Sensitivity Analysis),而且它能够结合先进的数值方法进行推进系统不同级别精度的仿真.用户可以在主界面上自由组装推进系统,在例库中我们给出了几种构造好的涡喷、涡扇推进系统模型. EPSP 目前最新的版本为 EPSP 1.0,该版本发布于 2005 年 9 月.

EPSP 是在综合研究了西方一些设计理念比较先进的推进系统仿真程序基础上开发出来的^[4-6].其实现环境为 Java,具有很强的可扩展性、易维护性和友好的用户界面.由于采用面向对象的建模方式,用类的继承的方法,可以很容易根据旧的推进系统部件模型设计出新的模型. EPSP 的拖放式用户界面非常友好,用户可以很容易掌握 EPSP 的操作.目前为止,EPSP 已经对某型涡扇发动机进行了初步仿真计算.

1 EPSP 主要特点

对于推进系统这样复杂的系统,传统的建模和仿真方法难以灵活的对其进行几何和物理表达, EPSP 的一个目标就是要能适应不同精度和不同学科的模式,它的内部数据结构要能提供对 CAD 网格生成几何学和可视化软件工具的直接访问,还要提供一种机制来耦合不同学科间和不

同保真度间的数据. 本系统主框架设计将通过 Java 面向对象编程语言来实现.

先进的推进系统仿真需要一个分布式仿真环境,它应该是可灵活定制的,可扩展的,并且用户界面友好的.它应当允许组件被其它类似功能的组件替换,这样用户可以自己定制仿真功能.这种“插件化”的设计方法对保持体系结构是很必要的,它能比较容易的支持与新仿真技术和新计算方法的结合. 本系统支持分布式计算,它将采用多个广泛使用的软件分布机制,例如 CORBA 和 RMI.

对象组合为将推进系统的物理结构表现出来提供了强大的机制.从结构的角度看,推进系统都是由进气道、风扇、压气机、燃烧室、外涵道、涡轮、轴以及喷管等部件组成.组合也可以递推的应用于表现模型的分级结构,例如我们考虑风扇部件,风扇的物理结构可以分解成:轮毂,轮盖,级以及流场,这些子部件中每个都可以在软件模型中表示成一个对象,它们合起来形成了风扇组合对象;而级对象,可以进一步分解成转子叶片和静子叶片对象.

仿真复杂系统需要模型分级或者多模型,代表系统处于不同深度的抽象之中.每个特定模型的选择都基于数个标准(可能有抵触),包括所需细节的精度、仿真的目的、可用的数据以及程序所能实现的功能.在推进系统的初步设计中,仿真模型经常用于求推进系统的推力、燃料消耗等等.这类性能预测一般用简单的零维部件模型来仿真实现.但是,在其它场合,尤其是多学科的分析中,可能需要更高等级的模型.例如,为了防止风扇叶片摩擦飞机引擎罩,工程师往往进行叶片的空气动

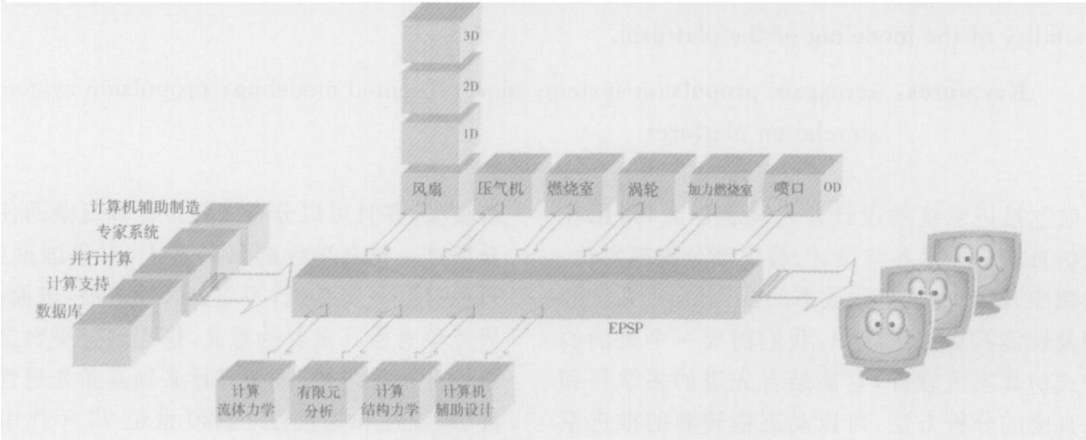


图 1 EPSP 框架结构
Fig. 1 EPSP's infrastructure

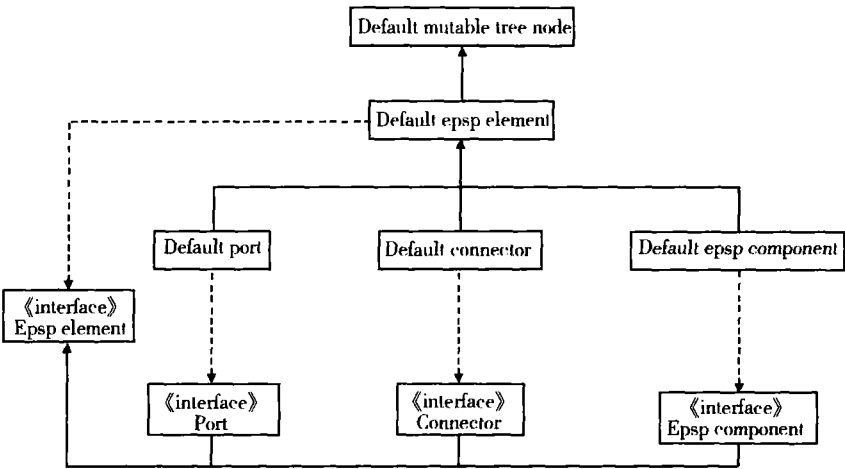


图 2 EPSP 物理模型 java 类结构关系图
Fig. 2 EpspElement java class structure diagram

力学, 热力学以及结构的分析, 以测定由于热和空气动力负荷而引起的叶片弯曲度. 这样的分析会需要多个用全三维、Navier-Stokes CFD 以及 FEA 算法的高精度模型. 图 1 为 EPSP 的结构框架图, 其中, 0 维为不考虑尺度分布的部件综合特性(总性能); 1 维为考虑轴向尺度的部件特性(沿程分布); 2 维为考虑轴向和径向尺度的部件特性(轴对称分布); 3 维为考虑轴向、径向和周向尺度的部件特性.

2 推进系统部件结构的工程模型

EPSP 的工程模型分为物理模型和数学模型, 物理模型是由 EpspElement 的对象来表示的, 包括推进系统部件、端口以及连接器的物理模型, 分别是由 EpspComponent, Port 以及 Connector 的对象来表示. 数学模型则是通过 ElementModel 的对象来表示.

考虑涡扇发动机中的进气道和风扇部件, 通过分析控制容积, 两个控制容积的边界共享进气道和风扇间的共同边界(或控制表面). 物质和能量就是通过这个面在进气道和风扇之间交换的. 这个面我们就抽象称为端口, 部件的行为如何确定呢? 通过守恒定律可以产生一组方程, 可以描述发生在控制容积中的物理过程. 就气动热力学的分析而言, 一般可以建立连续性、动量以及能量的守恒关系式.

EpspElement、EpspComponent、Port、Connector 以及 ElementModel 在 EPSP 中都被定义成 Java 接口. Java 接口不包含代码, 它定义了一系列空的方法, 而实现这个接口的 java 类必须为

这些方法定义具体的行为, Java 接口为描述设计意图, 而不是局限于特殊实现提供了强大的机制.

2.1 物理模型的基本元素(EpspElement)

EpspElement 是组建推进系统物理模型的基本元素, 它抽象了推进系统部件模型, 端口以及连接器的基本特征.

EpspElement 是一个接口, DefaultEpspElement 是这个接口的具体实现, 同时它还继承了 DefaultMutableTreeNode 类. DefaultMutableTreeNode 有一个 TreeNode 的引用, 它代表父节点, 以及一系列代表子孙节点的 TreeNode, 很好的满足了模型局部-整体分层结构的要求.

EpspComponent, Port, Connector 分别继承了 EpspElement, 用来表示系统的物理部件模型, 端口和连接器. 图 2 所示为物理模型的 java 类结构图.

同时, EpspElement 还包含对它的 ElementModel 对象的引用. 这些对象的引用通过变量名 model 指明, 注意这些变量引用的也是 Java 接口, 这样 EpspElement 实际上引用的是实现那些接口的对象, 这样就提供了插件化的模型, 同时也为模型的组合提供基础.

2.1.1 部件的物理模型(EpspComponent)

EpspComponent 对象用于代表物理控制容积, 描述具体的推进系统子部件, 比如压气机, 涡轮等等. DefaultEpspComponent 继承 DefaultEpspElement 类, 同时实现 EpspComponent 接口,

EpspComponent 也包含了对相应的 ElementModel 的引用, 例如压气机(Compressor)默认引用的数学模型是 DefaultCompressorModel,

涡轮 (Turbine) 默认引用的是 DefaultTurbineModel, 这些数学模型的类都继承了 DefaultComponentModel, 也就实现了 ElementModel 接口。

不同的部件之间会有很多共同的特征, 例如压气机和涡轮都属于叶轮机械, 同时它们又有各自的特点. 面向对象机制为此提供了极好的解决方案, 面向对象的三个基本原则是: 封装性、继承性和多态性. 封装性强化了代码的可读性和可维护性, 继承用于在多个部件类型中找出共同点, 提炼成抽象的部件类, 防止重复编码, 同时提高代码可维护性, 例如 GasTurbo 抽象类代表了整合压气机, 涡轮共有功能的抽象父类. 图 3 展示了 EPSP 的部件物理模型类的详细继承关系。

由于继承了 DefaultEpspElement 类, 因此部件的物理模型类也是局部-整体分层结构的, 一般来说单个部件类的子孙节点就是它的端口, 可以通过 getChilderen() 得到, 而多个部件以及它们之间的连接器也可以组合成一个综合部件, 例如高压压气机、燃烧室、高压涡轮以及它们之间的连接器合起来可以组成核心机。

2. 1. 2 端口 (Port)

Port 代表了控制容积的一个表面, 通过它流过一些实体 (如质量或能量). DefaultPort 继承 DefaultEpspElement 类, 同时实现 Port 接口, 与 EpspComponent 类似, Port 也包含了对 DefaultPortModel 类的引用, EPSP 中定义了四种类型的端口: GasPort、FuelPort、ShaftPort 和 ForcePort。

2. 1. 3 连接器 (Connector)

相连接的两个控制容积间的接口用连接器 (Connector) 对象来表示, 因为这个接口代表了两个控制表面间信息 (如质量, 能量等等) 的交换, 它在本系统中实际就定义成两个端口 (Port) 间的连接. 同时连接器还定义了方向, 表示流体流动或机械传动的方向, 这样整个推进系统的流路就非常清晰了. DefaultConnector 是 Connector 的默认实现, 具有代表来源 Port 和目标 Port 的引用。

2. 2 推进系统元素的数学模型 (ElementModel)

ElementModel 表示推进系统元素的数学模型, 它也是树型结构的, DefaultElementModel 实现了 ElementModel 接口, DefaultComponentModel, DefaultPortModel, DefaultConnectorModel 分别表示部件, 端口以及连接器的数学模型, 它们都继承了 DefaultElementModel 类, 其中最为复杂的是部件的数学模型。

推进系统部件行为是用数学方程定义的. 一般来说, 这些方程及其伴随的边界条件太复杂了, 根本不可能找到分析解法, 通过几何网格上的方程离散化可以得到近似的解法, ElementModel 封装了数学方程、网格图形和解算程序。

ElementModel 的复杂性使它从 EpspElement 中独立出来进行封装, 这也强化了 EpspElement 的模块化, 使新的 EpspElement 类不用考虑 ElementModel 就可以添加进来, 相反, 添加新的 ElementModel 也不会影响 EpspElement 类; 而且, 将数学模型封装到一个单独的对象里, 这种设

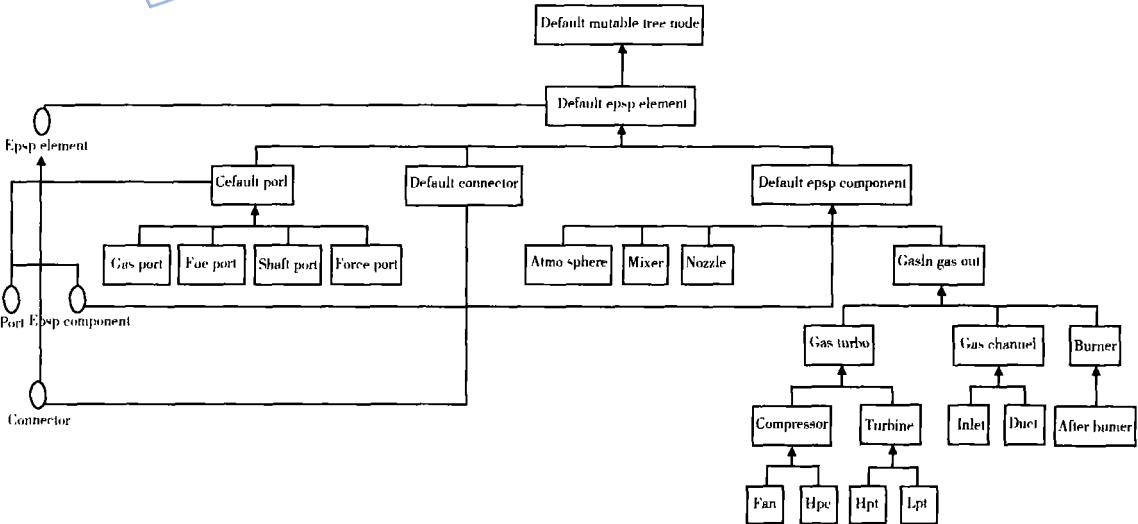


图 3 EPSP 物理模型详细类结构图

Fig. 3 Detailed EpspElement class structure diagram

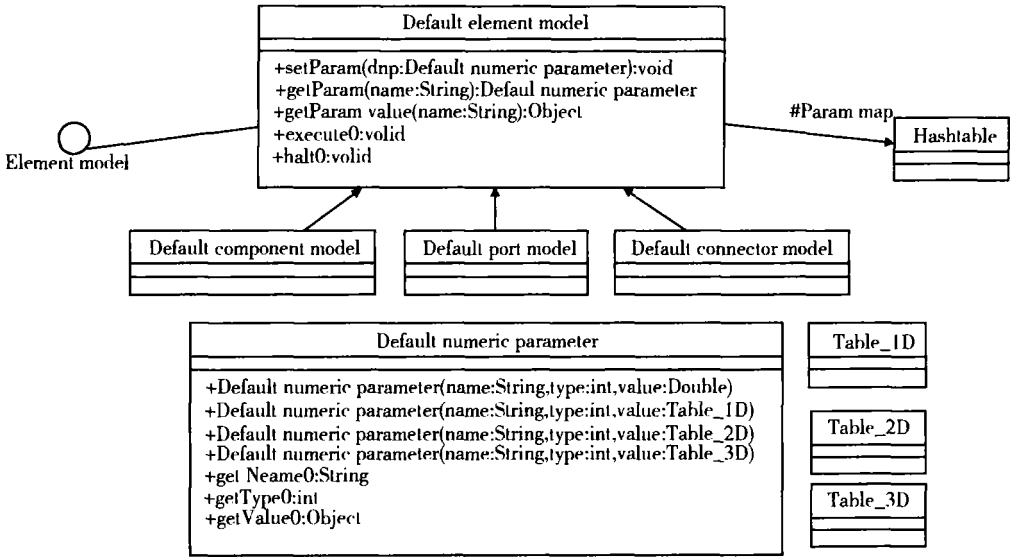


图 4 EPSP 数学模型框架类图

Fig. 4 Element Model class structure diagram

计可以实现对已经有的外软件包的“捆绑”。

DefaultElementModel 类实现了 ElementModel 接口, 图 4 是 ElementModel 的一个简单继承结构图, DefaultComponentModel、DefaultPortModel 以及 DefaultConnectorModel 分别代表部件、端口以及连接器的数学模型。对应于部件或端口的继承结构, DefaultComponentModel 类或者 DefaultPortModel 类会有相对应的子类, 例如 Fan 类, 相对应的就是 DefaultFanModel 类。而且, 按照学科和精度, 部件数学模型子类会通过继承产生一系列的子孙类。例如: 对于 Fan 的数学模型, 按学科来分, DefaultFanModel 可以被继承产生 AeroFanModel 类和 StructuralFanModel 类, 再进一步按照精度级别来分, 又可分别产生 ZeroDAeroFanModel 类、OneDAeroFanModel 类、ZeroDStructuralFanModel 类、OneDStructuralFanModel 类等等。

为了解决推进系统全通流部分的三维数值模拟的需求与计算资源不足之间的矛盾, 作为一种过渡性的手段, zooming 技术应运而生。zooming 技术也称为不同复杂程度分析技术 (Variable Complexity Analysis), 例如它可以把对整个推进系统通流部分的低阶精度模型 (零维或一维) 的计算与局部某一部件内的高阶精度模型 (三维) 的计算结合起来。DefaultConnectorModel 类中定义方法 transform() 来进行部件间的 zooming。

ElementModel 中封装了一系列的参数, 但是这些参数可能是 0 维的, 即仅仅是一个数, 用一个

double 型变量就可以表示, 也可以是 1 维、2 维, 或者 3 维的, 为此, 本系统分别定义了三个类: Table_1D, Table_2D, 以及 Table_3D 来存放对应的数据, 这些数可以在建立类的对象时直接进行初始化, 也可以读取一个数据文件, 例如风扇的特性数据就可以从一个指定的文件中读取, 每个类中还定义了相应的插值计算函数。

另外每个参数的类型也不一样, 例如有的参数是输入型的, 需要用户指定, 有的参数则是部件本身的固有属性, 需要预先给定初值, 用户不能更改, 而还有的参数则是经过计算得出来的, 我们构造了类 DefaultNumericParameter, 构造函数的定义是:

```
public DefaultNumericParameter ( String
name, int type, Object value);
```

其中 name 是参数的名称, type 是参数的类型, value 则是此参数具体的值, 如果是 0 维, 就是 Double 类的对象, 1 维、2 维、3 维则分别是 Table_1D、Table_2D、Table_3D 的对象。如果该变量尚未赋值, 那么 value 就是 null。

ElementModel 中提供了一个存放参数的 java.util. Hashtable 类的实例 paramMap, Hashtable 是一个可增长的键值一一对应的列表, paramMap 中存放的则是参数名称和该参数对应的 DefaultNumericParameter 实例。

3 一个实例

这部分我们用一个例子来说明本系统建模的

过程,在本文的第二部分,将会通过可视化的界面来引导用户进行这些工作.发动机被分成进气道,风扇、压气机、燃烧室、涡轮等等,我们这里以风扇为例,说明 EPSP 建模的过程.

3.1 物理模型的实现

先定义一个 GasPort 类如下:

```
public class GasPort extends DefaultPort{
    然后定义了一个 GasPort 类的构造函数:
    public GasPort()
    {
        this.name = "Gas Port";
        this.model = new DefaultGasPortModel
    ();
    }
}
```

类似的,定义 FuelPort, ShaftPort 和 ForcerPort.

接下来定义 Fan 类,它是 Compressor 类的子类:

```
public class Fan extends Compressor {
    然后定义了一个 Fan 类的构造函数,这个函数设置风扇的名称,并且创建了端口对象,如下所示:
    public Fan()
    {
        this.name = "Fan";
        inport = new GasPort ("fluid in");
        outport = new GasPort ("fluid out");
        shaftport = new ShaftPort("shaft in");
        addPort(inport);
        addPort(outport);
        addPort(shaftport);
        this.model = new DefaultFanModel();
    }
}
```

3.2 数学模型的实现

上面已经建立了风扇及相关端口的物理模型类,这里要建立相关的数学模型的类,假设采用默认的 DefaultFanModel 类、DefaultGasPortModel 类等等,它们分别继承了 DefaultComponentModel 类和 DefaultPortModel 类.在这些类的构造函数中,共同的一点就是先设定它们包含的参数,例如 DefaultGasPortModel 中可以设置参数如下:

```
setParam ( new DefaultNumericParameter
("temperature", 0, new Double( 288) ));
```

```
setParam ( new DefaultNumericParameter
("design temperature", 0, new Double( 288)
));
```

另外,DefaultFanModel 类中封装了风扇的具体的行为(用数学公式表示),通过 excute 方法调用自身以及端口中的参数进行运算,计算结果可以保存到 Port 对象中,通过连接器传递到相邻部件的 Port 中,为计算新的部件做准备.

如果要进行风扇的一维气动计算,风扇的数学模型类采用 OneDAeroFanModel 类代替 DefaultFanModel 类,相应地,端口数学模型类也要进行变更,用户可以在风扇的数学模型的树型结构列表中方便的选择,本文的第二部分将会进一步描述.由于部件的物理模型和数学模型是独立的,所以风扇的物理模型不受影响.

这个示例说明我们可以利用 EPSP 很方便的架构推进系统模型,可以根据不同的仿真要求,灵活的改变某部件的数学模型,这也正体现了“插件化”的思想.

4 总 结

推进系统仿真在从单一的、定制的仿真程序,向多学科综合的、适应各种需要的仿真环境发展.为了适应这种发展的需要,国内外都进行了燃气轮机的分布式仿真环境的实现研究.本文作者开发了基于 java 的面向对象的推进系统仿真平台,本文描述了此平台中所用的工程模型结构设计思路.工程模型为 EPSP 中推进系统仿真的基于组件的模型开发提供了基础,推进系统的各物理模型元素可以可视化的表现出来,它们的数学模型可以分布在远程机器上,这些将在本文的第二部分进行讨论.

参考文献:

[1] Reed J A, Afjeh A A. Computational simulation of gas turbines: part I- foundations of component based models[R]. International Gas Turbine & Aeroengine Congress & Exhibition, 1999.

[2] 唐海龙,张津. 面向对象的航空发动机性能仿真程序的设计方法研究[J]. 航空动力学报, 1999, 14(4): 421-424.

TANG Hailong, ZHANG Jing. A study of object oriented approach for aeroengine performance simulation[J]. Journal of Aerospace Power, 1999, 14(4): 421-424.

[3] 陈策,王文栋,李军. 面向对象的发动机特性计算的建模与仿真[J]. 机械科学与技术, 2003, 7 月(增刊): 34-36.

CHEN Ce, WANG Wen dong, LI Jun. An object oriented modeling and simulation of turbofan characteristic[J]. Me

chanical Science and Technology, 2003, 7: 34-36.

[4] Visser W P J, Broomhead M J. A generic object oriented gas turbine simulation environment[R]. ASME Turbo Expo 2000, Munich, Germany, 8-11 May 2000.

[5] Reed J A, Afje A A. Computational simulation of gas turbines[R]. In: International gas turbine & Aeroengine Congress & Exhibition, Indianapolis, Indiana, USA, June 7-10, 1999, 1-11.

[6] Bayt R L, Sahm M K, Lents C E. Collaborative engineering in integrated aircraft power system design [R]. ASME 2002 Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Montreal, Canada, September 29-October 2, 2002.

