

文章编号: 1000-8055(2007)02-0268-06

可扩展的推进系统仿真平台: 部分 II-可扩展的框架结构

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. EPSP 提供了一个灵活的仿真环境, 可以定义、修改和仿真基于组件的推进系统模型. EPSP 具有友好的图形化界面, 可以方便地修改推进系统各部件的参数, 用户可以进行定制和扩展, 以增加新功能或者适应所需的仿真行为. 本文分别介绍了 EPSP 的可视化框架和连接服务框架, 并且研究了计算框架的设计思想. 文中给出了某双轴涡扇发动机动态仿真的算例, 结果表明, EPSP 是进行扩展灵活的推进系统仿真计算的有力工具.

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 II - extensible framework

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. EPSP provides a flexible environment for defining, modifying and simulating the component-based propulsion models. EPSP has a friendly GUI for easily modifying parameters of the components. EPSP enables users to customize and extend the framework to add new functions or adapt simulation behavior as required. The visual assembly framework and the connection services framework were introduced, and the design pattern of the computation framework was studied. An example of transient simulation of a twin-spool turbofan engine was provided. It shows that EPSP is effective in extendable and flexible simulation computation for propulsion system.

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

在推进系统研制、使用、维护各阶段必须进行大量的计算机仿真研究, 这些仿真活动涉及气动热力学、结构和材料科学、自动控制、决策科学等众多学科门类. 以往的研究已成功地将上述各仿真活动与大系统隔离开来, 并为其定义了清晰的边界条件和研究目的, 他们各自沿着不同的学术

路线发展, 相互之间虽有渗透融合, 但总的说仍处于分散、小规模、无组织的状态, 这给许多复杂现象的建模仿真带来了极大困难. 随着计算机软件工程和网络技术的长足进步, 依据系统科学规律将推进系统寿命周期内的各种独立的仿真活动综合成为一个有机整体的时机已日益成熟^[1].

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.

在综合研究了西方一些设计理念和先进的推进系统仿真程序^[2-4]基础上, 本文建立了可扩展的面向对象的推进系统仿真平台(Extensible Propulsion Simulation Platform, 简称 EPSP) .

本文主要描述了 EPSP 的体系结构. EPSP 提供了一个灵活可扩展的环境, 帮助用户很方便的组合新的推进系统模型, 检查修改已有的模型, 并且进行仿真并显示结果. 这些模型是基于本文第一部分中所描述的推进系统部件框架的, 它将推进系统部件模型封装成组件的形式, 提供了一种可插拔的机制, 用户可以组合很复杂的推进系统模型. 下面的部分将会展示 EPSP 平台的框架结构, 接下来的三部分分别描述各个框架的主要特征, 包括: 图形界面, 分布式连接服务以及计算. 本文最后给出了一个算例.

1 EPSP

EPSP 主要由 4 部分组成:

- 1) 推进系统部件框架. 这个框架结构包含了代表推进系统物理部件的软件集合, 部件是基于本文第一部分定义的工程模型来表现的.
- 2) 可视化框架. 可视化框架是以推进系统部件框架为基础的, 提供了部件框架中对象的可视化的表现形式.
- 3) 连接服务框架. 这个框架是为部件框架和可视化框架提供本地和分布式的连接服务.
- 4) 计算框架. 这个框架提供了一套 EPSP 进行仿真的工具, EPSP 定义了一套模型方程语言规范, 并提供了一系列解决方法.

这些框架的构建增强了系统的灵活性, 它们之间相对独立.

2 EPSP 可视化框架

工程上经常利用示意图来代表推进系统的系统和子系统, 可视化框架部分基于推进系统的部件模型, 提供了一个可以灵活创建, 集成的体系结构.

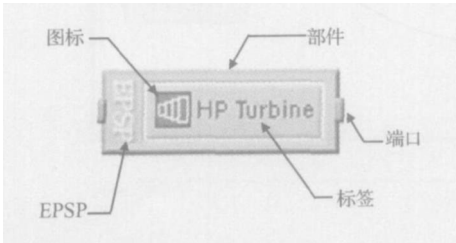


图 1 推进系统部件的可视化控件

Fig. 1 Schematic icon

2.1 可视化控件

如图 1 所示是一个推进系统部件的可视化控件, 代表一个推进系统物理部件, 它包含一个图标, 表示部件的图像, 标签表示所代表的部件的名称, 同时每个控件上都包含若干个端口, 代表部件间的连接点. 推进系统部件, 端口以及连接器的控件都包含弹出式菜单, 菜单中有些基本的操作, 例如: 放大、缩小等等, 每个部件都会有相应的树型 Element Model 结构, 也就是数学模型, 例如, 对于风扇 Fan, 就是一个以 DefaultFanModel 为根节点, 继承的子孙类为叶结点的树型结构, 用户首先要根据仿真要求的精度和学科来选择数学模型, 默认的则是 DefaultFanModel; 端口的数学模型则取决于所属部件的数学模型. 根据所选取的数学模型, 用户可以通过图形化用户界面对部件及其端口的状态和结构参数进行设置, 如图 2 为风扇用户界面, 通过它用户可以设置风扇的各种参数以及风扇所包含端口的参数; 同样地, 压气机、涡轮、燃烧室等等也都可以根据需要进行设定. 另外, 用户可以很方便的查看部件的特性图, 并进行特性数据的修改. 图 3 所示为风扇特性图的示例.

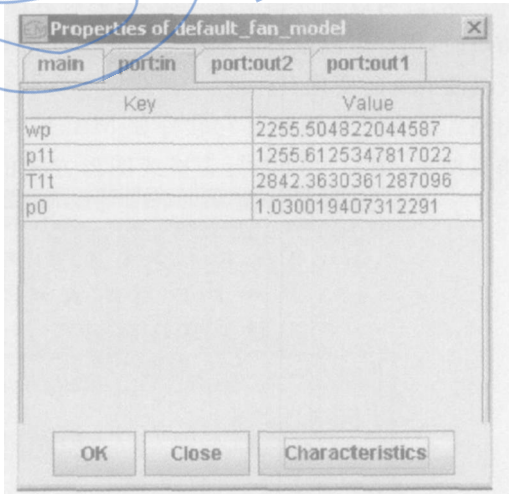


图 2 风扇的示例参数界面

Fig. 2 Fan's parameter interface

2.2 可视化建模界面

EPSP 基于组件结构开发的可视化模型为用户提供了一套界面友好的人机对话环境. EPSP 允许组件被其它类似功能的组件替换, 用户可以自己定制仿真功能, 能进行多种类型的推进系统仿真; 如图 4 所示, 推进系统部件模型采用可视化图标来表示, 多个部件图标按一定的排列组合起来就形成了一个有效的推进系统模型. 在模型窗口中

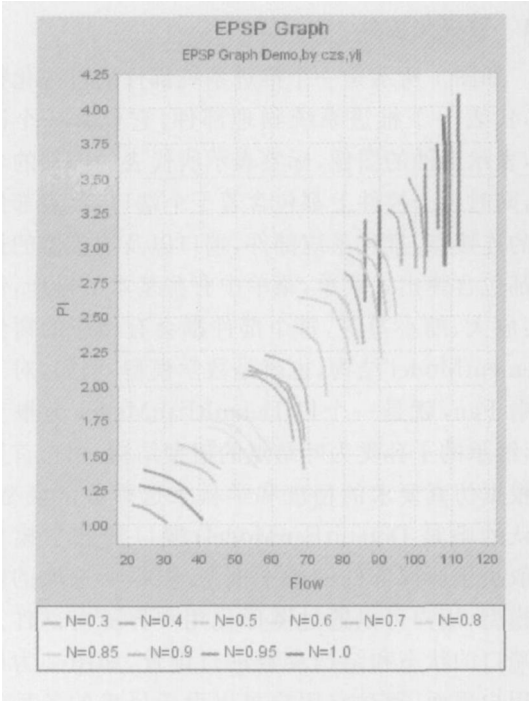


图 3 风扇的示例特性图

Fig.3 Fan's performance map

可以很方便的对部件图标进行拖放,也可以在模型间进行部件图标的拷贝,这样用户可以创建自己的部件仓库,并可以作为模型保存起来;另一方面,进行仿真时,用户可以很方便的制定各种仿真控制参数及其随时间的变化,还可以选择或定制各种数据输出格式以方便应用到其它软件如 Matlab 中,对于仿真结果,用户可以进行表格或图形方式的查

看,并且可以以文件的形式保存下来.

2.3 数据格式

推进系统部件结构的工程模型具体数据以及关联统一采用 XML 存放,XML 是结构化的、平台和语言无关的,并且具有高度的可扩展性,近年来已经飞速发展成不同应用程序间交换数据的新兴标准.

用户的各种定制也采用 XML 来保存,用户进行基于 web 的 EPSP 仿真时,用户的定制文件保存在用户的计算机中,仿真开始后,配置文件以 XML 的形式通过 Internet 传到 EPSP 仿真服务器上,服务器接收到配置文件后,自动把文件分发给仿真执行组件来进行仿真.

3 连接服务框架

由于单台微机的计算能力有限,而有强大计算能力的超级计算机又过于昂贵,目前很多的研究已经采用分布式并行计算.分布式对象技术主要是在分布式异构环境下建立应用系统框架和对象构件.在应用系统框架的支撑下,开发者可以将软件功能封装为更易管理和使用的对象,这些对象可以跨越不同的软、硬件平台进行互操作.目前,分布式互操作标准主要有 Microsoft 的 COM/DCOM 标准、Sun 公司的 Java RMI 标准和 OM G 组织的 CORBA 标准.

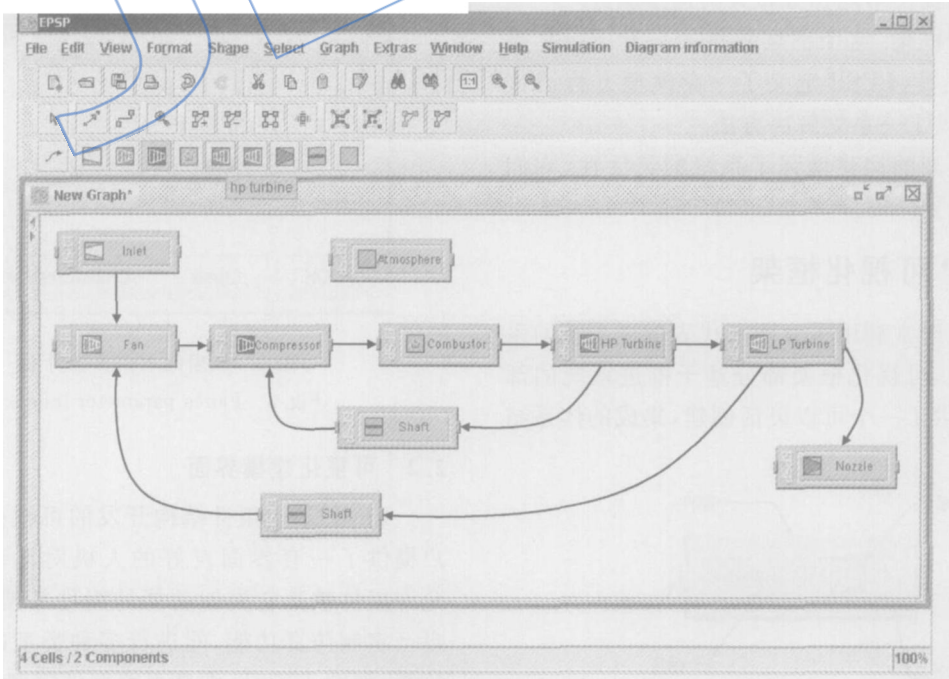


图 4 EPSP 主界面

Fig.4 Main graphic user interface

作为分布式应用程序框架的规范, COBRA 首当其冲, 它是由对象管理组织 (OMG) 开发的. CORBA 能够利用不同编程语言(例如 C/C++、Basic 等) 开发实现分布式应用, 而 RMI 是一种纯 Java 解决方案. 在 RMI 中, 程序的所有部分都由 Java 语言编写, 这样, 开发出来的程序完全符合 Java 规范, 便于实现跨平台访问、扩展和移植.

EPSP 连接框架的主要目标是为了开发一套统一的机制来实例化或者引用 ElementModel, 为了达到这个目的, 框架要求能兼容各种不同的分布式对象机制, 例如: RMI, Corba 等等, EPSP 中分布式机制的原理如下: ElementModels 被放在远程服务器上, 并通过不同的分布式对象机制(例如 RMI, CORBA) 等等, 连接到 EPSP. 推进系统部件的可视化控件的弹出式菜单中将可以选择连接到远程服务器上的分布式 ElementModel 对象, 当分布式连接建立好之后, EPSP 将会列出所选择的远程服务器上所有可用的 ElementModel. EPSP 的分布式机制目前已经实现了 RMI 的方式, CORBA 的支持不久也将完成.

Java RMI 机制允许一个线程无缝地调用一个远程对象的方法. 远程对象是指那些驻留在其他虚拟机上的对象, 因此远程对象既可以在同一台计算机上的不同虚拟机 Java 中, 也可以在通过网络连接的远程主机上的虚拟机中. 如图 5 所示, 系统由以下几个层次组成:

- (1) 代理层 (Stub/Skeleton Layer): 客户端存根 (Stub) 和服务端端框架 (Skeleton), 它们屏蔽了远程引用层和传输层的细节, 提供了 Java 应用程序调用远程对象方法的透明接口.
- (2) 远程引用层 (Remote Reference Layer): 远程引用语义、行为、细节.
- (3) 传输层 (Transport Layer): 连接的建立、管理, 远程对象的追踪.

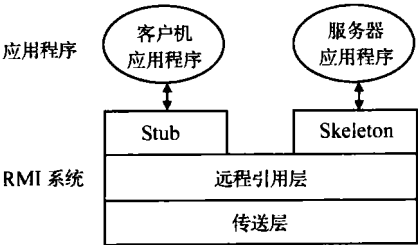


图 5 连接框架采用 RMI 的层次关系图
Fig. 5 RMI hierarchy diagram

4 计算框架

计算是进行仿真的关键步骤, 大多数情况下, 它需要为描述推进系统行为的一系列数学方程找到解法, 解决过程的本质取决于问题的复杂度, 定义方程的本质, 以及特殊边界条件, 初始值等等. EPSP 的计算任务主要是在 ElementModel 中来进行的, 在本文的第一部分中, 已经介绍了 ElementModel 可能是对已有外软件包的“捆绑”, 也可能自身包含用于求解的数学方程, 第一种情况下, EPSP 根据用户输入的初始参数以及边界条件, 调用外软件包进行计算, 显示计算结果并且根据用户的需求, 将结果由端口通过连接器传递给下一个需要计算的部件的端口中. 对于第二种情况, EPSP 有一套自定义的建模方程语言, 能求解系统中数学方程, 它主要被用于系统建模的各种连续性方程, 下面将重点介绍.

图 6 所示为计算框架的 java 类结构图, 在 ElementModel 中定义了以下两个方法:

- 1) void setExpressions (String expressions);
初始化 ElementModel 的时候调用这个方法, 来进行此 model 包含的数学表达式的初始化;
- 2) String getExpressions ();
表示取得 ElementModel 目前所有的数学表达式.

Expressions 的定义必须符合 EPSP 系统自定义的语法, 一个表达式就是一个公式, EPSP 支持算术、关系以及逻辑表达式. 算术表达式中的语法能代表任何类型的数学操作, 例如: “+”、“-”、“*”、“/” 分别代表加减乘除, “*” 代表幂运算. EPSP 也支持 IF、WHILE、FOR 等条件语句和循环语句.

本文第一部分中给出了 ElementModel 中变量的定义, 一个部件的 model 中的表达式是由 model 本身的变量、部件端口的 model 中的变量、中间变量以及运算符所构成, 格式如下所示:

$$WR = g_in. W * \text{sqrt}(g_in. T) / g_in. P \tag{1}$$

WR 为 model 中定义的某变量的名称, g_in 为 model 中端口的对象名, 调用端口 model 中的变量采用 portname. variable 的形式. 最终需要求解的问题实际上涉及多个部件、端口以及端口之间的连接器, 它们的 ElementModel 中的 expressions 组合起来, 形成方程组, 由 EPSP 求解. 用户界面上每个需要计算的部件都是 EpspElement

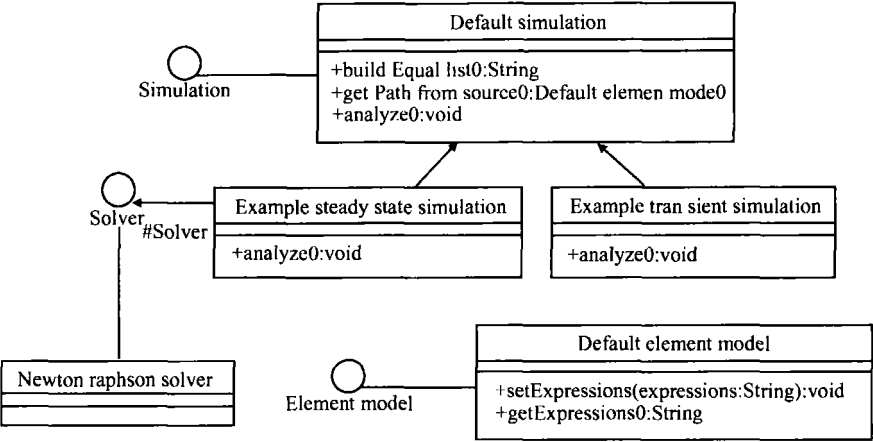


图 6 EPSP 计算框架类图

Fig. 6 Computation framework class structure diagram

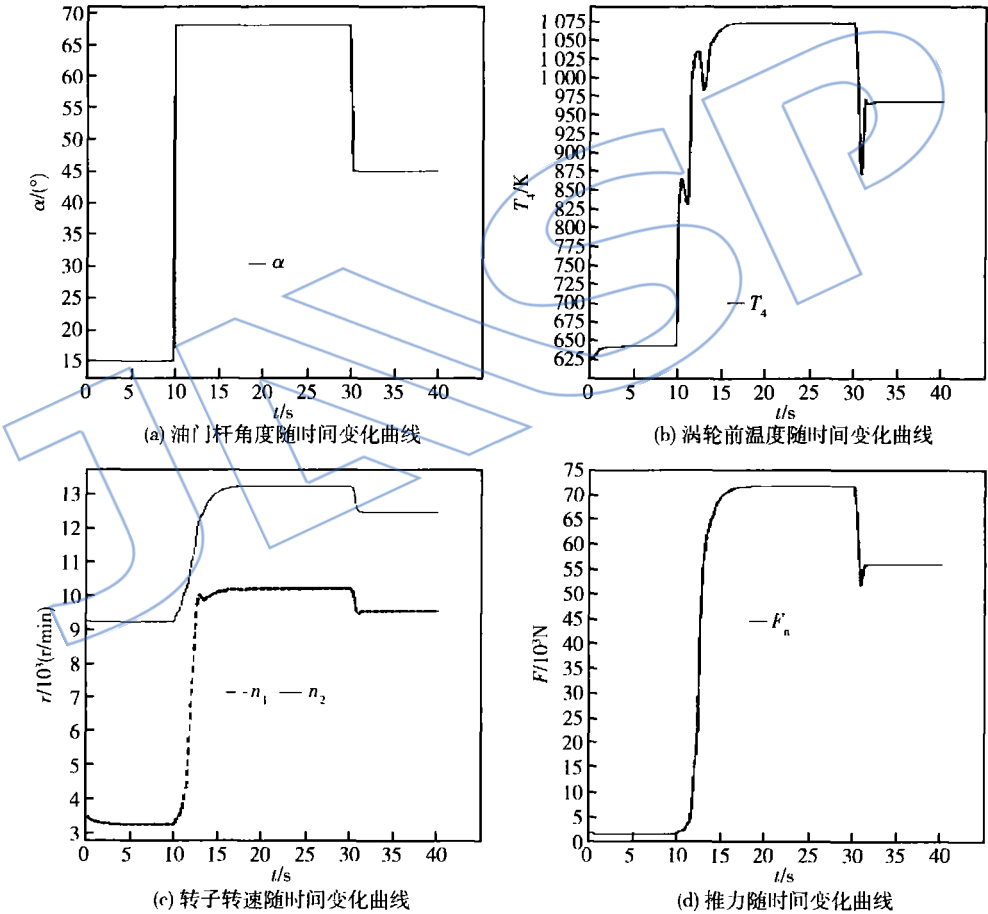


图 7 仿真结果

Fig. 7 Simulation results

的一个具体实例, 都有一个对象名, 假设 1) 式所在部件对象名是 *comname*, 则 (1) 式在求解时则转变成:

$$\begin{aligned} comname.WR &= comname.g_in.W * \\ &sqrt(comname.g_in.T) / comname.g_in.P \end{aligned} \tag{2}$$

另外 EPSP 还包含一套数学函数类库, 提供常用算法的支持, 它们都实现自 Solver 接口。例如 NewtonRaphsonSolver 类就封装了 NewtonRaphson 算法。

对于具体的推进系统的仿真问题, EPSP 定义了 Simulation 接口, 通过实现 Simulation 来封

装具体的仿真策略, Simulation 接口中定义了 analyze() 方法, 用于分析执行仿真策略, DefaultSimulation 类是 Simulation 接口的默认实现. 例如 ExampleSteadyStateSimulation 和 ExampleTransientSimulation 两个类分别继承了 DefaultSimulation 类, 用于模型的稳态和瞬态模拟. ExampleSteadyStateSimulation 中将 analyze() 方法具体化, 根据模型中部件的连接以及方向, 自动读取各个部件的方程并进行分析, 进行复杂方程的简化, 以及各个部件的方程间的排序, 并且检查方程之间的矛盾和冗余. 默认调用 NewtonRaphsonSolver 类来进行求解.

5 仿真实例

在标准大气条件下, 对某型涡扇推进系统进行地面台架状态仿真模拟. 油门杆角度变化过程如下: 0~ 10 s 时, 油门杆角度为 15°; 在第 10 s 时, 油门杆角度骤然拉升到 68°, 并保持到第 30 s; 第 30 s 时, 油门杆角度骤然降低到 45°, 并持续到第 40 s. 图 7 所示为 EPSP 的图形输出界面显示的部分参数随时间变化曲线.

6 展 望

参考国外优秀的设计思想, 我们开发出扩展性强的、灵活易用的推进系统系统仿真平台 EPSP. 目前的 EPSP 1.0 初步实现了一些框架性的

构建, 在此基础上, 我们将继续逐步完善整个平台的结构、部件模型以及整合各种先进的数值求解方法, 并在分布式并行计算、对其他已有的成熟软件(包括商业软件)进行封装等方面取得进一步提高, 还将引入数据库系统存放各种类型的数据, 如用户资料, 推进系统及各部件的各种参数, 试验数据等等. 力求能尽快参与到推进系统的可行性预测、设计、改进、故障诊断分析以及教学等方面工作中.

参考文献:

[1] 谢志武, 苏明, 翁史烈. 燃气轮机系统仿真集成概念及其分布对象实现[J]. 航空动力学报, 2000, 15(3): 307-310.
XIE Zhiwu, SU Ming, WENG Shilie. Gas turbine simulation integration and its distributed object implementation [J]. Journal of Aerospace Power, 2000, 15(3): 307-310.

[2] 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.

[3] Reed J A, Afjeh A A. Computational simulation of gas turbines: part II-extensible domain framework [R]. International Gas Turbine & Aeroengine Congress & Exhibition, 1999.

[4] Visser W P J, Broomhead M J. GSP: A generic object oriented gas turbine simulation environment [R]. ASME Turbo Expo 2000, Munich, 2000.