

# 基于 Python 开发 PSS/E 高级应用程序

廖晓晖, 张沛超

(上海交通大学电气工程系, 上海 200240)

**摘要:** PSS/E 是 Siemens PTI 公司研制的支持二次开发的电力系统仿真软件, PSS/E 30 版本增加了对 Python 语言的支持。首先介绍了 Python 作为一种“胶水语言”的工作机制, 然后阐述了如何利用 Python 和面向对象的方法建立电网模型, 并在此电网模型的基础上实现 PSS/E 的各种高级应用的方法。论文以电网孤立岛的判断和线路定步长等高级故障分析为例, 详细说明了面向对象的 PSS/E 高级应用程序的开发过程和特点。通过算例证明, 利用 Python 开发 PSS/E 的高级应用程序具有结构开放、开发周期短、易于扩展等优点, 特别适合于开发大型、复杂 PSS/E 高级应用。

**关键词:** PSS/E; Python; 孤立岛; 面向对象

## Developing advanced PSS/E applications based on Python

LIAO Xiao-hui, ZHANG Pei-chao

(Department of Electrical Engineering, Shanghai Jiaotong University, Shanghai 200240, China)

**Abstract:** PSS/E is a power system simulation software developed by the Siemens PTI, which provides a powerful library of application programming interface so that researchers can develop advanced applications. The PSS/E 30 now support the object-oriented programming language called Python. This paper first introduces the characteristics of Python. Then it illustrates how to build the power grid models using the Python language and applying the object-oriented method. After that, the paper discusses in detail the process as well as the characteristics of developing object-oriented advanced applications based on PSS/E. An island-detection program and a line-step-fault program are employed to demonstrate the feasibility and advantages of the method proposed in this paper. The results shows that Python will play an important role in developing large, complex applications based on PSS/E.

**Key words:** PSS/E; Python; isolated island; object-oriented

中图分类号: TM743 文献标识码: A 文章编号: 1003-4897(2008)11-0009-04

## 0 引言

就二次开发能力而言, 电力系统离线仿真软件的发展经历了几个阶段。最初的软件几乎不提供用户自定义功能, 操作方式为全人机交互方式, 如 EMTP、BPA 等。随着对电力系统仿真分析要求的不断提高, 在仿真过程中经常要进行大量重复性计算。例如, 在保护整定计算中需要计算分支系数, 在电网静态安全分析中需要进行“N-1”开断分析。若依旧采用人机交互方式则效率非常低下。故现代的电力系统仿真软件开始向研究人员提供二次开发接口, 允许研究人员开发脚本程序, 以便扩充模型、自动完成大批量的仿真计算及结果分析<sup>[1-4]</sup>。PSS/E 是这类分析工具的代表。PSS/E 为用户提供和支持用于二次开发工作的编程语言: IPLAN 语言, 以及

丰富的应用程序编程接口 API (Application Programming Interface) 函数<sup>[5,6]</sup>, 使得该软件更能满足工程用户的实际需求。作者已基于 IPLAN 语言开发了大量高级故障分析软件, 在实际中发挥了重要作用。但是, IPLAN 语言仍然存在两个较大问题: (1) IPLAN 是传统的结构化编程语言, 不支持面向对象开发。当用于大型电力系统分析软件开发时, 开发效率较低, 且代码难以维护; (2) 更重要的是, IPLAN 不是通用语言, 它无法与其他广泛采用的编程语言 (如 C、C++ 等) 集成, 也不支持数据库、网络等应用。因此, 它是一种为 PSS/E 所独有的、封闭的开发语言。

在 PSS/E30 版本中, 支持 Python 开发语言, 这标志着电力系统仿真软件的二次开发正向着面向对象、通用以及开放的方向发展。本文首先对 Python

语言做简要介绍, 然后分别从电网模型、网络拓扑分析以及高级故障分析等几个方面, 说明如何利用 Python 开发面向对象的 PSS/E 应用程序。

### 1 Python 语言

现代电力系统仿真软件的功能已不仅仅局限于分析计算, 还要求支持面向对象建模, 支持数据库技术, 支持网络应用等。为此, 迫切需要一种具有开放结构的“胶水语言”, 可以将电力系统仿真分析、数据库、网络以及图形界面的最新技术成果集成起来。

Python 是一种近几年流行起来的跨平台的、面向对象的、通用的、开放的编程语言。作为脚本语言, Python 的设计目标之一就是成为一种“胶水”语言。在基于 PSS/E 开发高级应用时, 其作用如图 1 所示。首先, Python 能够直接调用 PSS/E 内部提供的丰富的 API 函数, 从而完全控制 PSS/E 以实现各种短路、潮流计算功能; 其次, PSS/E 能方便的调用其他通用语言(如 C/C++)编写的程序库。只需要写少量的包装类, 就可以在 Python 中调用 C/C++ 程序库, 这使得扩充 Python 的功能非常方便, 不需因改用 Python 而完全放弃之前用 C/C++ 所做的工作, 而只需用 Python 写“胶水代码”即可。最后, Python 还支持各种数据库应用和网络应用, 从而实现计算参数的高效存储管理以及数据的传输与共享。

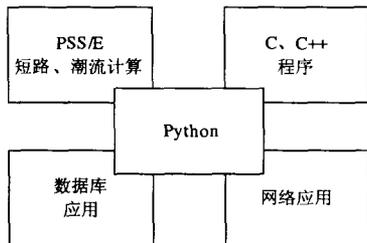


图 1 Python 是一种“胶水语言”

Fig.1 Python is a kind of “Glue Language”

### 2 电网模型

支持面向对象的系统建模和开发是 Python 的首要特征。根据 PSS/E 中参数的格式, 应用统一建模语言 UML (Unified Modeling Language) 可以建立电网模型, 如图 2 所示。

下面仅以母线为例, 说明如何用 Python 语言描述母线的属性。

```
class bus:      #定义母线类
    NumberOfBus=0          #电网母线数
    def __init__(self, BusID, BusName, BusType, BaseKV,
```

```
GB):          #母线构造函数
    self.BusID=BusID      #母线编号
    self.BusName=BusName  #母线名称
    self.BusType=BusType  #母线类型
    self.BaseKV=BaseKV    #基准电压
    self.GB=GB            #对地导纳
    NumberOfBus+=1
```

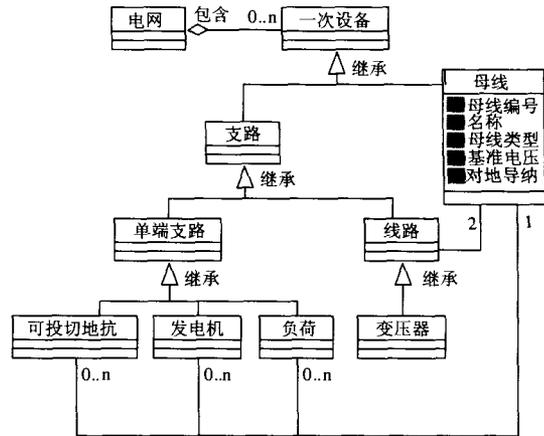


图 2 电网 UML 模型

Fig.2 UML model of the power grid

### 3 网络拓扑分析

网络拓扑分析是进行各种电网分析计算的基础。在建立了电网的面向对象模型的基础上, 可以进行面向对象的网络拓扑分析。本文以孤立岛判别为例, 说明如何进行网络拓扑分析。

#### 3.1 孤立岛判别的重要性

在利用 PSS/E 的电网故障计算分析或者网络等值等功能之前, 需要确保当前的电网模型是一个连通的拓扑结构, 否则容易出现异常。而不连通即孤立岛的情况经常出现, 主要原因是线路或者变压器的运行状态发生变化。比如图 3 所示的电网结构原本是连通的, 但是如果在运行过程中, 将 102\_103\_1, 102\_105\_1 等两条线路做临时开断, 则会出现母线 102 与该网络其他部分成不连通的状态。此时再调用 PSS/E 进行故障分析或者潮流计算时, 则可能会出现异常。

因此, 在利用 PSS/E 进行各种故障分析或潮流计算之前, 对当前电网的状态进行一次连通性检查是很有必要的。根据面向对象的分析方法, 可以得到如图 4 所示模型图。图 4 中, 孤立岛判别器封装了孤立岛诊断算法, 其输入是面向对象的电网模型, 输出是判断结果。

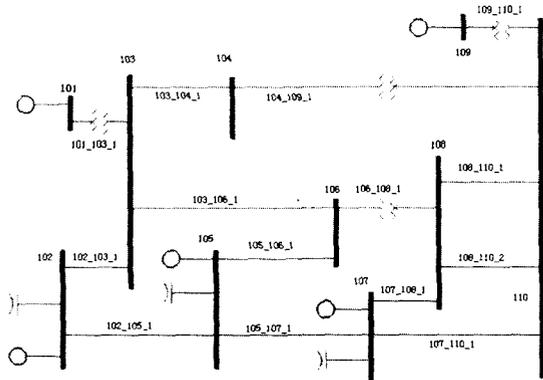


图3 示例网络图

Fig.3 An example power grid

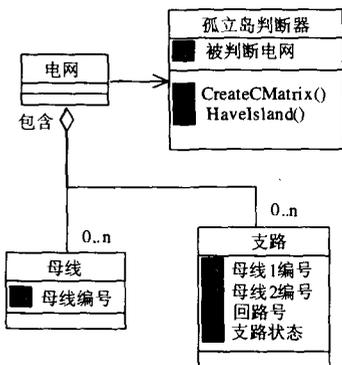


图4 孤岛判断UML模型

Fig.4 The UML model of island detection

### 3.2 孤岛判别算法

以图4为例, 来介绍如何进行电网连通性的判断。结合 PSS/E 数据模型以母线为中心的特点, 将孤岛的判断分为两步, 即根据电网模型生成相伴矩阵  $C$  (CreateCMatrix()) 和判断该网络是否连通即是否有孤岛存在 (HaveIsland())。

孤岛判断器之 CreateCMatrix() 函数首先生成一个  $n \times n$  的相伴矩阵  $C$ , 其中  $n$  为电网中母线的数量,  $c_{ij}$  由式(1)决定:

$$c_{ij} = \begin{cases} 0 & \text{母线}i\text{与母线}j\text{之间无连通的支路} \\ 1 & \text{母线}i\text{与母线}j\text{之间有连通的支路} \end{cases} \quad (1)$$

孤岛判断器之 HaveIsland () 则根据生成的相伴矩阵  $C$ , 得到能达矩阵  $R^{[9]}$ :

$$R = (C + I)^{n-1} \quad (2)$$

其中:  $I$  为  $n$  阶单位矩阵。当矩阵  $R$  的所有元素均为 1 时, 该矩阵所描述的电网拓扑结构连通, 否则说明含有 0 的行所对应的母线与网络其他母线不连通。

### 3.3 电网连通性判断算例

对于图3所示的网络, 当网络中所有元件均正

常投运时, 网络的相伴矩阵  $C_1$ , 如此可以得到能达矩阵  $R_1$ , 而当线路 102\_103\_1 与线路 102\_105\_1 停运时, 母线 102 以及该母线上所带的其他元件一起, 将形成孤立岛, 可得此时的网络的相伴矩阵为  $C_2$ , 能达矩阵为  $R_2$ , 通过  $R_2$  可知, 网络中母线 102 及所带元件与电网中其它元件孤立形成孤立岛。

$$C_1 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} \quad R_1 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (3)$$

$$C_2 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} \quad R_2 = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

## 4 面向对象的故障分析

基于 PSS/E 的电网故障计算分析功能可分为两类: 一类是基本或简单故障分析功能, 即通过人机交互一次可以完成的功能; 另一类为高级故障分析功能, 即由多个基本功能以一定的规律组合在一起实现的功能, 比如轮断计算、零序电流计算、分支系数计算、线路定步长故障分析、阻抗定步长分析等。图5是利用 Python 所实现的 PSS/E 故障分析对象模型。其中高级故障类将复杂的故障分析转变成一系列的简单故障, 之后由简单故障对象生成 PSS/E 故障计算脚本, 提交给 PSS/E 计算并最后取得计算结果。

通过以上分析可知, 若需扩展新的故障分析功能, 只需在高级故障类下面派生新的故障分析类即可。

下面以较为简单的定步长计算为例, 阐述如何利用 Python 实现 PSS/E 高级故障分析功能。如图3所示的电网, 101\_106\_1 的 A 相从线路 5% 处到 95% 处以步长 5% 发生单相接地故障, 过渡阻抗为  $10 \Omega$ 。利用 Python 实现的线路定步长高级故障分析类的定义如下:

```
class LineStepFault: #定义线路定步长故障计算类
def __init__(self, LineName, FaultType,
```

```

FaultPhase, BeginPos, EndPos, Step, RX, xjRX, JumpType)
self.LineName=LineName #故障线路名称
self.FaultType=FaultType #故障类型
self.FaultPhase=FaultPhase #故障相
self.BeginPos=BeginPos #定步长起始位置
self.EndPos=EndPos #定步长结束位置
self.Step=step #步进长度
self.RX=RX #接地过渡阻抗
    
```

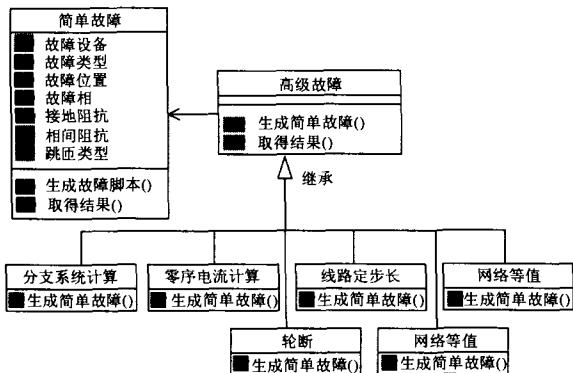


图5 高级故障分析UML模型

Fig.5 The UML model for advanced fault analysis

```

self.xjRX=RX #相间过渡阻抗
def GenerateSingleFaultList(self) #生成简单故障列表
def GetCalculateResult(self) #获得计算结果
    
```

然后，线路定步长的计算过程如下：

```

AlineStep=LineStepFault(101_106_1,AGA, 0.5,0.95,0.05,10,0) #定义线路定步长对象
SingleFaultList=AlineStep.GenerateSingleFaultList() #生成简单故障列表
for AFault in SingleFaultList:
    AFault.Calculate() #简单故障计算
    AFault.GetResult() #取得简单故障计算结果
    LineStepFault.GetCalculateResult() #取得定步长计算结果
    
```

以上是线路定步长计算的顶层代码，可以看出，利用 Python 进行 PSS/E 的高级应用程序开发方便、灵活而且易于扩展。

### 5 总结

电力系统仿真软件往往具有专用、封闭的特点。而利用 Python 语言，不但实现了仿真分析系统的面向对象建模和实现，而且在专用软件与通用开发语言、通用的数据库、网络技术之间建立了桥梁，从而极大增强了电力系统仿真分析系统的功能和实用性。作者成功开发了大量故障分析高级应用，证明了本文方法的可行性与有效性。

限于篇幅，本文仅选取电网面向对象建模、拓扑分析以及故障计算为例，说明如何利用 Python 开发 PSS/E 高级应用。而如何利用 Python 进一步集成基于其他语言（如 C、C++、JAVA 等）开发的分析

计算库，以及如何与数据库、网络等应用进行集成，则不在本文讨论范围，读者可进一步参阅文献[10]。

### 参考文献

- [1] 罗中远,张沛超,郁惟镛,等. 基于PSS/E27的大型电力系统故障分析平台[J]. 继电器,2004,32(9):57-59. LUO Zhong-yuan, ZHANG Pei-chao, YU Wei-yong, et al.Fault Analysis Platform of Large Power System Based on PSS/E 27[J]. Relay, 2004,32(9):57-59.
- [2] 聂宇本,高翔,张沛超. 继电保护日常运行管理专家系统的开发[J]. 继电器,2002,30(9):50-52. NIE Yu-ben,GAO Xiang,ZHANG Pei-chao.Development About Expert System of Daily Performance Management for Relay Protection[J]. Relay, 2002,30(9):50-52.
- [3] 王威,郁惟镛,张沛超. 面向对象的继电保护整定计算专家系统的研究[J]. 电力系统及其自动化学报,2001,13(2): 1-4. WANG Wei,YU Wei-yong,ZHANG Pei-chao.Research on Object-Oriented Expert System of Relay Setting Calculation[J].Proceedings of the CSU-EPSA,2001, 13(2): 1-4.
- [4] 张扬,陆承宇,杜振东. 电力系统仿真软件PSS/E简介[J]. 华东电力,1997,(1):6-9. ZHANG Yang,LU Cheng-yu,DU Zhen-dong. Introduction of PSS/E[J]. East China Electric Power, 1997,(1):6-9.
- [5] Power Technologies Inc(PTI).PSS/ E - 30 Program Application Guide Online Documentation[Z].2004.
- [6] Power Technologies Inc(PTI).PSS/ E - 30 Program Operation Manual Online Documentation[Z].2004.
- [7] 国家电力调度通信中心.电力系统继电保护规定汇编[M].北京:中国电力出版社,2000. State Electric Power Dispatching Center of China. Compilation of Power System Relay Protection Regulation[M].Beijing: China Electric Power Press,2000.
- [8] 张沛超,郁惟镛,李翔,等.应用UML开发智能继电保护的整定计算程序[J].电网技术, 2001,25(4): 32-35,39. ZHANG Pei-chao,YU Wei-yong,LI Xiang,et al.Application of Unified Modeling Language to Development of Intelligent Setting System for Protective Relaying[J].Power System Technology, 2001,25(4):32-35,39.
- [9] 储俊杰. 变电所一次主接线电气连通性分析的数学模型[J]. 电力系统自动化,2003,27(1): 31-33. CHU Jun-jie. Mathematical Model for Analysing Electrical Connectedness of Main Electrical Scheme in Substation[J]. Automation of Electric Power Systems, 2003,27(1): 31-33.
- [10] Hammond M, Robinson A. Python Programming on Win32. O'Reilly Media/Inc. 2000.

收稿日期：2007-09-29； 修回日期：2007-12-04

#### 作者简介：

廖晓晖(1983-)，男，硕士，研究方向为电力系统继电保护整定计算理论及相关软件技术研究；E-mail:liao\_xh@sjtu.edu.cn

张沛超(1970-)，男，博士，副教授，主要研究方向为继电保护、电网调度自动化以及人工智能。