**工作流模型**

—基础软件部

摘要

本文主要从概念上介绍了基础平台中工作流模型，工作流中有哪些业务对象，业务对象间的相互关系，业务对象具有哪些活动，并以请假业务为例，介绍了绘制流程图的主要工作，即梳理业务活动及其环节要素。

目录

[一、工作流、工作流模型概念 2](#_Toc486608652)

[二、业务对象概念 3](#_Toc486608653)

[1.结构部分 3](#_Toc486608654)

[1.1流程类别（WORKFLOW\_TYPE） 3](#_Toc486608655)

[1.2流程(WORKFLOW) 4](#_Toc486608656)

[1.3流程变量（WORKFLOW\_VARIABLE） 5](#_Toc486608657)

[1.4环节（NODE） 5](#_Toc486608658)

[1.5办理环节（TRANSACT\_NODE） 6](#_Toc486608659)

[1.6活动环节（ACTIVITY\_NODE） 6](#_Toc486608660)

[1.7嵌套环节（NESTED\_NODE） 8](#_Toc486608661)

[1.8复合环节（COMPOSITE\_NODE） 9](#_Toc486608662)

[1.9决策环节（DECISION\_NODE） 9](#_Toc486608663)

[1.10开始环节（START\_NODE） 9](#_Toc486608664)

[1.11结束环节（END\_NODE） 10](#_Toc486608665)

[1.12页面（PAGE） 10](#_Toc486608666)

[1.13流程和页面关系（WORKFLOW\_PAGE） 10](#_Toc486608667)

[1.14环节和页面关系(NODE\_PAGE) 11](#_Toc486608668)

[1.15流向（ROUTER） 12](#_Toc486608669)

[2.实例部分 12](#_Toc486608670)

[2.1流程实例（WORKFLOW\_INSTANCE） 12](#_Toc486608671)

[2.2流程变量实例（WORKFLOW\_VARIABLE\_INSTANCE） 13](#_Toc486608672)

[2.3环节实例（NODE\_INSTANCE） 14](#_Toc486608673)

[2.4任务实例（TASK） 14](#_Toc486608674)

[2.5任务页面实例（TASK\_PAGE\_INSTANCE） 16](#_Toc486608675)

[三、业务对象关系 16](#_Toc486608676)

[四、业务对象活动 17](#_Toc486608677)

[1.启动 17](#_Toc486608678)

[2.签收 17](#_Toc486608679)

[3.提交 18](#_Toc486608680)

[4.退回 18](#_Toc486608681)

[5.撤回 18](#_Toc486608682)

[6.委办 18](#_Toc486608683)

[7.删除 18](#_Toc486608684)

[8.特送退回 19](#_Toc486608685)

[五、绘制流程图 19](#_Toc486608686)

## 一、工作流、工作流模型概念

首先，什么是工作流？

工作流是由一系列有先后顺序，并且需要多人协作的业务活动组成，其本质是业务活动流。如请假流程，请假流程中至少有申请、审批这两项业务活动，业务活动顺序必须是先申请再审批，并且申请、审批这两项业务活动由不同的人完成，如图1。

图1 请假流程

工作流的运行依赖于两部分，一部分为结构部分，另一部分为实例部分。

结构部分指流程的主体结构（如图1），以流程图的方式描述，流程结构包括环节与流向，环节对应业务活动，而流向决定了业务活动运行的顺序、方向。

实例部分指流程实际运行时的方式，结构与实例的关系类似于Java中类与对象的关系，流程会产生流程实例，环节会产生环节实例，实例部分由工作流引擎运行时产生。如张三启动请假流程时，会产生张三请假流程这个流程实例。

工作流的概念明确后，那么什么是工作流模型？

工作流模型是基础平台提供的一种功能，它的使用需要基于平台中的组织机构模型和角色权限模型，该模型包括流程类别管理（如图2）、流程设计（如图3）两个模块。流程类别管理模块将不同的业务流程进行分类，便于管理；流程设计模块提供了可视化流程设计器来绘制业务流程图。



图2 流程类别管理

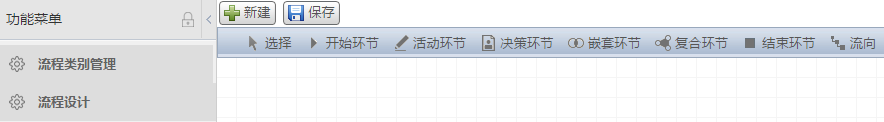


图3 流程设计

## 二、业务对象概念

工作流中业务对象可分为结构部分、实例部分。

## 1.结构部分

## 1.1流程类别（WORKFLOW\_TYPE）

把流程按一定业务概念进行分类则产生了流程类别。流程类别本身为一树形结构树。它只是在逻辑上划分流程，以方便浏览的作用,如图2中的文件夹。

**关系：**@上级类别，+@流程

**属性：**

* ID
* 名称：类别名称
* 上级类别：本类别的所属上级类别
* 描述：流程类别的描述信息

**功能**：在流程类别管理模块中，提供了新增、修改、删除流程类别的功能

[新增]：点击，新增加一个流程类别

[修改]：点击，修改一个流程类别的名称、上级类别

[删除]：点击，删除一个流程类别，当该流程类别包含子流程类别或流程时不可删除

## 1.2流程(WORKFLOW)

　　指流程结构部分，其中记录一个流程的全部信息，包括环节与流向。

每个流程必须拥有版本号，每个流程默认的版本号为1。当一个流程已经发起过流程后（产生流程实例），不能直接修改原流程，而是应该利用新增+版本号+创建时间+创建人的方法，来解决流程修改的问题。

流程中的环节必须有开始环节和结束环节，开始环节只可以有一个，结束环节可以有多个。

**关系：**@类别，+@环节， +@流程变量，+@页面，+@流程实例

**属性：**

* ID
* 版本：同一个流程的不同版本只能有一个被发起
* 流程编码：区分不同业务流程
* 名称：流程名称
* 描述：流程的描述
* 流程启动者：流程的发起人可以是一个具体的人，也可以是一个嵌套环节实例。
* 流程类别：该流程所属的流程类别
* 流程处理优先级别：分别为高、中、低三种级别。流程的优先级别影响到每个用户待办事宜的排序，优先级别高的流程的待办任务会排列靠前。
* 前处理程序：流程启动前所运行的程序，程序可由sql语句或存储过程表示
* 后处理程序：流程结束后所运行的程序，程序可由sql语句或存储过程或Java表示
* 业务状态字典：为流程配置一个业务状态字典，可在流程中的不同环节选择使用

**功能：**在流程设计模块中，提供了新建流程、绘制流程图、保存流程的功能

[新增]：点击，新增一个流程。

[绘制流程图]：利用流程设计器绘制流程图，绘制步骤将在文末作进一步介绍。

[保存]：点击，将保存流程的所有结构部分的信息，包括环节与流向。

[删除]：点击，删除一个流程。如果已经产生了流程实例，则不能被删除。

## 1.3流程变量（WORKFLOW\_VARIABLE）

流程变量指事先在流程中定义好的变量及其初始值，可在流程运行过程中用作控制流程分支方向。

**关系：**@流程

**属性：**

* ID
* 名称：变量名称
* 流程：变量所属的流程
* 初始值：变量创建时的默认值

## 1.4环节（NODE）

环节可按照不同的分类标准分为不同的类型。如不同的业务活动产生了不同的环节类型，如办理环节、活动环节、决策环节、嵌套环节、复合环节；不同的环节位置产生不同的环节类型，如开始环节、结束环节；这里的环节指各种类型环节的超类，其中保存了各类型环节共有的基本属性、通用属性，其它的各种类型环节全部继承于该环节产生。

**关系：**@流程，+@页面，+@环节实例

**属性：**

* ID
* 名称：环节名称
* 坐标X、坐标Y：环节坐标，绘制流程时使用
* 描述：环节描述信息
* 类别：开始环节、结束环节、活动环节、决策环节、嵌套环节、复合环节
* 业务状态：流程运行到该环节时对应的业务状态，通过业务状态字典选择
* 流程：环节所属流程

## 1.5办理环节（TRANSACT\_NODE）

　　指需要用户进行办理的环节，继承自环节。

　　办理环节中拥有办理人的属性，办理人实际配置为一个角色，流程运行过程中，会根据这个角色获得真正的任务办理人。

**属性：**

* 环节办理人:流程执行过程中的实际办理人由角色产生
* 前处理程序: 环节实例化时所运行的程序，可由sql语句或存储过程表示
* 后处理程序: 环节实例完成后所运行的程序，程序可由sql语句或存储过程或Java表示

## 1.6活动环节（ACTIVITY\_NODE）

指工作流程中的一个基本的业务活动环节，需要用户办理，继承自办理环节，所以拥有办理人这个属性。同时它也拥有一些表示流程处理逻辑的特别属性。

一个活动环节可以配置多个办理页面，这些页面是这个环节办理人所需要办理任务的载体，即任务的办理是通过填写活动环节的表单完成的。这些页面是在流程设计时注册的，在环节中直接选择使用。针对于每个不同环节的页面可能会有不同的配置，如页面是否可查看，是否可办理，是否可审批等。页面中具体的控件受制于页面本身的状态，如对于只读状态的页面，那么其内控件的状态也只能在只读、隐藏中选择，而不能选择编辑。

**属性：**

* 多人办理方式：
* 抢占式

指如果某活动环节虽然是多人办理，但谁先签收，则成为任务的办理人，其他办理人员无法再签收。

* 会签式

指如果某活动环节是多人办理，那么每个人都成为任务的处理人，此时根据会签参数的设置来决定任务办理的结果。

* 会签处理参数
* 全部人员

指这个环节中的所有办理人员全部完成，环节才完成。

* 比例人员

指环节中的所有处理人员在固定的百分比人员完成后，环节就完成，其他人员无需再办理。

* 固定人员

指指定数量的人员完成后，环节就完成，其他人员无需再办理。

* 会签值：

在会签处理参数为比例人员或固定人员时，这里填写的是人员百分比或固定人员数。

* 是否聚合：
* 不聚合：流程同一时间只选择一个分支运行

当流程经过决策环节，从上分支或下分支其中一个分支运行时，环节6不聚合。

图4



* 聚合：流程同一时间选择多个分支运行

当流程从上分支、下分支两个分支同时运行时，环节6聚合。

图5



* 不同意跳转至环节：当退回时，设置退回到的环节

## 1.7嵌套环节（NESTED\_NODE）

　　继承于环节。绑定子流程，主业务流程执行到在嵌套环节时，产生多个子业务流程，主流程与子流程业务上差异大，且有依赖关系，如上下级关系，如：警卫处某工作流程需要下发到各个分县局办理，警卫处工作的完成依赖于各个分县局工作的完成，警卫处业务为主流程，各分县局业务为子流程，在警卫处主流程中有一个“分县局办理”的嵌套环节。嵌套环节绑定的子流程会产生流程实例。

**关系：**@流程

**属性：**

* 子流程：环节绑定的流程
* 默认退回节点：退回子流程时，默认退回到的子流程中的节点

## 1.8复合环节（COMPOSITE\_NODE）

继承于环节。绑定子流程，主业务流程执行到在复合环节时，并没有产生子业务流程，而是继续依次执行子流程中包含的若干环节，将若干环节组织为一个子流程并绑定到主流程中复合环节的目的在于，可以使得主流程不用关注子流程中各环节具体执行过程，主流程只需关注复合环节本身。实际上，复合环节只是以流程的方式组织了若干环节并将该流程绑定于自身，复合环节只是为了绑定那若干环节。使用复合环节可以实现将若干环节合并展示为一个环节的前端效果。复合环节绑定的子流程不会产生流程实例，仍然使用了主流程的流程实例。

**关系：**@流程

**属性：**

* 子流程：环节绑定的流程
* 默认退回节点：退回子流程时，默认退回到的子流程中的节点

## 1.9决策环节（DECISION\_NODE）

继承自办理环节。它的出现代表后续的环节需要判断流程走向，并且这个判断只会有一个条件成立，即流程只会选择后续一个分支走向运行。判断的依据为流向上设置的条件，一般使用流程变量赋值表达式作为决策条件。

**属性：**

* 决策类型：

自动决策

当可根据业务数据明确流程走向时，为自动决策。如请假流程中，可根据请假天数决定是否需要经理审批环节。

手动决策

当根据业务数据无法由程序明确判断流程走向时，由人员手动决策。

## 1.10开始环节（START\_NODE）

开始环节代表流程的开始，一个流程中只能有一个开始环节，代表只能从一个环节开始流程。开始环节只能有一个流出流向。

## 1.11结束环节（END\_NODE）

结束环节代表流程的结束，一个流程中可以有多个结束环节，代表流程可以从多个分支结束。结束环节可以有多个流入流向。

## 1.12页面（PAGE）

用户办理工作的界面是页面，用户办理工作就是在页面上填写数据。页面上可以挂载表单、列表、图片等多种元素, 根据页面上挂载的元素不同，页面可分为多种类型。当完成一个页面时，将该页面交给平台的页面管理模块进行管理，即将页面信息注册到页面管理模块，页面管理模块中的页面可以被工作流选择使用。

**关系：**+@流程，+@环节

**属性：**

* ID
* 名称：页面名称
* 编码：页面编码
* 类型：表单、列表、签章、提醒
* 模块：页面所属模块
* URL：指页面在项目中的相对路径，与表单唯一对应，作为页面的属性被管理在页面管理模块。

## 1.13流程和页面关系（WORKFLOW\_PAGE）

使用工作流之前需要把所有相关页面配置到工作流中，配置时可从页面管理模块中直接选择已完成的页面，为工作流配置页面的过程就是在建立流程与页面间的对应关系。一个流程可以对应多个页面，一个页面也可以运用于多个流程。

工作流中使用到的页面一般为表单类型（form）。

**关系：**@流程，@页面

**属性：**

* ID
* 名称：流程页面名称
* 页面：页面管理模块中的页面
* 流程：所属流程

## 1.14环节和页面关系(NODE\_PAGE)

环节页面关系指环节与页面间的对应关系。环节页面关系是基于流程页面关系上而产生的，即为环节配置页面时，只能在该流程已配置的页面中选择。一个环节可以对应多个页面，一个页面也可以运用于多个环节。

**关系：**@环节，@页面

**属性：**

* ID
* 标题：环节页面标题
* 页面：页面管理模块中选择的页面
* 环节：所在办理环节
* 控制状态：查看、办理、审批、签章（个性业务需求）

查看

查看指在这个环节表单仅可以查看表单，不能进行编辑修改操作。

办理

办理指在这个环节可以对表单进行查看、编辑操作，但不能处理表单意见区。

审批

审批指在这个环节表单可以进行查看、编辑操作，还可以添加审批意见。

## 1.15流向（ROUTER）

　　指流程中各环节之间的连接符，依据流向决定环节的后续环节及前续环节。

　　流向中可以编写条件，一般在决策环节后的流向才需要编写条件，根据条件决定工作流分支走向哪个环节。

**关系：**@流程，@环节

**属性：**

* Id
* 名称：流向名称
* 分支条件：

指分支条件用表达式描述，在实例运行中，由系统根据所书写的表达式来自动判断走向。

* 上一环节：流入环节
* 下一环节：流出环节
* 流程：所属流程

## 2.实例部分

流程的运行是以实例的方式进行的，流程实例运行的过程就是每个活动环节办理业务活动的过程。

## 2.1流程实例（WORKFLOW\_INSTANCE）

　　每启动一次流程会产生一个流程实例，一个流程可以被多个人发起，即产生多个流程实例，这个过程称为流程实例化。如请假这个流程，当张三需要请假时，张三发起请假流程，这时就产生了一个请假流程实例；当李四需要请假时，李四发起请假流程，这时又产生了一个请假流程实例。同一流程下的不同流程实例，可以通过不同的流程实例标题来区分，如上面例子中的两个流程实例可以区分为：张三请假流程、李四请假流程。

**关系：**@流程，+@环节实例，+@任务实例，～状态

**属性：**

* ID
* 启动人:可以是具体的用户，也可以是嵌套环节实例，由启动类型决定
* 启动类型：工作流的发起者可以是实际的用户，也可以是某一环节
* 人工启动

指人为手工的启动流程，最为常见的启动方式，如在页面中点击按钮启动

* 嵌套启动

指该流程由嵌套环节自动启动

* 状态：指在工作流运行过程中流程实例所处的状态
* 未提交状态

当一个业务流程被发起，尚未被提交到后续环节办理时为未提交状态。

* 运行状态

当一个业务流程被发起，且被提交到后续环节办理，但尚有环节未办理完成时，该流程实例为运行状态。

* 完成状态

当一个业务流程的所有环节业务活动全部办理完成后，那么该流程实例状态为完成状态。

* 流程：所属流程
* 标题：这个流程具体某个流程实例的业务描述。

如请假流程，每个人都可以启动，那么就会产生多个流程实例，可以采用“请假人+流程名称”作为实例标题加以区分，如张三请假流程、李四请假流程。

## 2.2流程变量实例（WORKFLOW\_VARIABLE\_INSTANCE）

流程变量实例指流程实例在运行的过程中，对于在流程设计中定义的流程变量的应用。当一个流程启动后会产生流程实例，同时初始化了流程变量实例。流程变量实例也可以在服务层通过java接口进行设置。

**关系：**@变量，@流程实例

**属性：**

* ID
* 变量：流程变量
* 流程实例：所属流程实例
* 变量值：变量值，可在环节时赋予

## 2.3环节实例（NODE\_INSTANCE）

　　启动一次流程后，产生一个流程实例，在流程实例运行的过程中，每经过一个环节就会产生一个环节实例，一个流程有多个环节，所以一个流程实例在运行过程中会产生多个环节实例，产生环节实例的过程称为环节实例化。如张三请假这个流程实例，在请假申请、组长审批、经理审批这几个环节都会产生环节实例。 决策环节只决定流程的走向，不会产生环节实例。

**关系：**@环节，@流程实例，+@任务实例，～状态

**属性：**

* ID
* 状态：指在工作流运行过程中环节实例所处的状态
* 运行

当前环节的业务活动尚未完成

* 完成

当前环节的业务活动已完成

* 流程实例：所属流程实例
* 环节：所属环节
* 完成时间：当前环节的业务活动完成时的时间

## 2.4任务实例（TASK）

启动一次流程，产生一个流程实例，当该流程实例运行到请假申请环节时，需要办理这项业务活动，而业务活动一定是由具体的人来办理的，因此在环节实例化的过程中结合环节办理人产生了任务实例，简称任务。办理一项业务活动，最终被描述为由具体的人来办理任务。

由于一项业务活动可能需要多人办理，所以一个活动环节在环节实例化时可能会产生多个人的任务。一个任务只能由一个人办理，一个人可以办理多个任务。

**关系：**@流程实例，@环节实例，@办理人，+@任务页面实例，～任务状态，

**属性：**

* ID
* 办理人：由环节办理人角色产生的具体每个任务办理人
* 派发时间：指任务产生（创建）的时间
* 签收时间：指办理人接受任务的时间
* 完成时间：指办理人完成任务的时间
* 环节实例：所在环节实例
* 流程实例：所在流程实例
* 任务状态：指在工作流运行过程中任务所处的状态
* 待办

任务产生时的初始状态

* 在办

任务已被签收，正在办理

* 已办

任务已办理，被提交、退回、委办

* 办理动作：用户办理任务所做的操作，任务的办理动作会导致任务状态发生相应的变化
* 无动作

任务办理人未做任何操作，任务状态为待办

* 签收

任务办理人确认签收任务，任务状态变为在办

* 提交

任务办理人正向（同意）办理任务，任务状态变为已办

* 退回

任务办理人逆向（不同意）办理任务，任务状态变为已办

* 撤回

任务办理人撤回已办理的任务

* 委办

任务办理人将任务委托给他人办理

* 是否完成：任务是否完成的标志

## 2.5任务页面实例（TASK\_PAGE\_INSTANCE）

任务需要由人来办理，而人办理任务就是在页面上填写数据，办理一个任务可能需要填写多个页面，一个任务可能对应多个任务页面实例，一个任务页面实例就是一个任务与一个页面的组合。

**关系：**@任务

**属性：**

* ID
* 任务：所属任务实例
* 临时数据：页面上填写的数据，供保存草稿使用
* 页面：完成任务所需的页面

## 三、业务对象关系

1.流程类别——流程（1..\*）

2.流程——环节（1..\*）

3.流程——流向（1..\*）

4.流程——流程变量（1..\*）

5.流程——流程实例（1..\*）

6.流程——页面（\*..\*）

7.环节——流向（\*..\*）

8.环节——环节实例（1..\*）

9.环节——页面（\*..\*）

10.流程实例——环节实例（1..\*）

11.流程实例——任务实例（1..\*）

12.环节实例——任务实例（1..\*）

13.任务实例——办理人（\*..1）

图6

## 四、业务活动

## 1.启动

　　启动是针对具体流程的操作。当用户需要办理一项业务时，他需要启动对应的业务流程，用户启动流程的过程，称为启动流程。每启动一次流程即产生一个该流程的实例。如张三需要请假，就需要启动请假流程，启动请假流程后会产生一个请假流程实例。

## 2.签收

签收动作是针对个人任务的操作。当在一个业务活动环节产生任务后，任务办理人需要签收该任务后再进行办理，办理人签收任务后任务状态变为在办，签收前任务状态为待办。

## 3.提交

　　提交动作是针对个人任务的操作。当用户接收到了一个业务流程产生的任务时，可以执行提交操作，提交操作代表同意办理，使流程往下一环节方向运行。

## 4.退回

　　退回动作是针对个人任务的操作。当用户接收到了一个业务流程产生的任务时，可以执行退回操作，退回操作代表不同意退回，使流程往上一环节方向运行。

## 5.撤回

撤回动作是针对个人任务的操作。撤回只有在任务办理完成后才可以触发，指把已经办理的任务撤回来重新办理。如张三填写了请假单并提交给领导审批，但是在提交后发现请假天数填写错了，这时张三就可以执行撤回操作把已经办理的任务撤回来重新填写。需要注意，撤回的前提是在下一环节办理人没有签收的情况下才能执行。

## 6.委办

委办动作是针对个人任务的操作。指个人任务可以由办理人委托给其他人处理。如当组长李四出差时，可以委托王五来审批张三的请假单。

## 7.删除

删除动作是针对流程实例的操作。如当张三发起请假流程，产生了一个请假流程实例后，此时张三发现不需要请假了，可以执行删除操作来删除该流程实例，同时会删除相关环节实例、任务实例。需要注意，删除操作一般只能由流程发起人执行。

## 8.特送退回

特送退回是针对个人任务的操作。指流程跳过中间环节，而直接退回到某一指定的环节。

图7

如上图所示，在一般情况下，流程如果需要退回到活动1环节，需要从活动4到活动3到活动2这样逐个环节退回。但是如果执行了特送退回操作，那么活动4可以直接退回至活动1环节办理。

在退回时，只能选择已办理的环节退回，不能选择未办理的环节来退回。

## 五、绘制流程图

基于上述内容可以了解到，要想使用工作流，先得有流程图。绘制流程图的关键在于梳理清楚流程中的业务活动，这里以请假流程为例分以下几个步骤介绍：

**1.梳理业务活动，对应为流程中的环节**

请假流程中，主要的业务活动可分为请假申请、请假审批，再结合具体的组织机构岗位，可将审批分为组长审批、科长审批、处长审批、经理审批等。

**2.梳理业务活动的先后顺序**

由业务活动本身的含义来梳理其先后顺序，请假流程中的业务活动，一定是先提出申请，再由领导审批。

**3.梳理流程的分支走向、决策条件**

如图1，当请假人请假天数在两天以内时由组长审批即可，当请假人请假天数超过两天时需要由经理审批。这个流程中有两个分支走向，决策条件是请假天数是否超过两天。

**4.梳理环节各要素**

* 办理人：请假申请环节的办理人可以是请假人自己，也可以是他人代为请假，组长审批环节的办理人是组长，而组长可能有多个人担任，因此办理人可以使用角色来表示。角色是基础平台中角色与权限模型中的概念，这里可以简单理解为一类人、多个人。
* 办理方式：当环节需要多人办理时，需要确定是抢占式，还是会签式。二者的区别在前文中已有介绍。
* 表单：请假流程使用请假单，包含请假人、请假时间、请假事由等信息。一个业务活动可能需要填写多个表单。
* 业务状态：根据实际办理进度可分为已申请、组长审批中、经理审批中、完成等状态。
* 是否可退回：在组长审批、经理审批环节可以退回，请假申请环节不可以退回。可以退回的环节一般具有审批权限。
* 前后置程序：主要用于完成流程自身的个性业务，或者驱动另一个业务流程，或更新其它相关业务对象的某些状态属性。对于请假流程，并没有类似的需求。

**5.绘制流程图**

依据上述分析梳理的结果，使用流程设计器绘制出流程图。流程图中流程、环节、流向的详细配置将在另一篇文档（工作流使用说明文档）中介绍。