BPMX4软件需求规格说明书

广州宏天软件有限公司

**文档版本记录表**

| 版本 | 日期 | 作者 | 更改情况描述 |
| --- | --- | --- | --- |
| V1.0 | 2013-05-20 | Winston | 创建 |
| V1.1 | 2013-05-29 | Winston | 完成第一个版本 |
|  |  |  |  |

# 文档概述

## 目的和背景

BPMX4是基于BPMX3进行开发，为了解决BPMX3的更灵活的业务扩展要求、以及增加更多的功能。

X4将从技术架构、系统模块、流程业务发布过程、组件化等角度来做重大的重构和改造。

## 定义

本文档汇集BPMX4项目的所有需求，对每项需求做简要说明，从业务范畴限定其范围。软件工程后续的工作，包括概要设计、详细设计等，均依据此文档进行设计。

## 术语

* 流程定义（Process-Definition）

又叫业务流程、工作流程。是反映业务目的工作流模型，这个模型定义了要实现的目标以及实现目标所要经过的过程。流程可以用流程图来表示。

该流程图通过支持bpmn2.0规范的规定性文档进行描述。流程定义由工作流引擎解释执行。

* 子流程（Sub-Process）

子流程不可以直接启动。子流程将作为工作流程的一个活动来执行。

* 数据源

存储数据的集合，目前支持关系数据库中的表，当数据源与流程绑定后，则数据源用来保存工作流运行过程中的数据。

* 活动Activity

活动是构成流程的基本单元，主要反映完成业务流程需要执行的操作。

包含的属性有：名称、描述、类型(人工任务、自动活动、子流程等)、是否起始活动、是否添加活动、执行人、活动时限、通知方式以及相应的表单信息等。

* 起始活动

表示一个业务流程的起点。当起始活动启动后，系统将自动生成一个流程实例。每一个流程最小有一个起始活动。起始活动目前是由人工来启动的。区别与别的工作流系统，起始活动本身也是一个活动，人工起始活动将启动表单，提供人机交互界面，与实际业务过程中的起草、申请、录入等工作相对应。在流程图中的活动方框内有一个明显的箭头图示表示起始活动，非起始活动则表示为齿轮。

* 人工任务

专指需要业务人员进行一些手工处理的活动。人工任务在在执行时将启动表单界 面，提供人机交互界面。

活动是最高层的抽象，任务继承活动，一般有人工任务（多种）、脚本任务、消息任务等。

* 自动任务

指无需人工干预，系统自动运行的活动。

* 添加活动

在人工活动时，其启动的表单界面是空白表单。而当非添加活动时，其表单为上一活动传递而来，当起始活动为非添加活动时，启动的表单将从已存在的表单中选取。一般情况下，起始活动同时也是添加活动。非起始活动同时也是非添加活动。只有一些特殊的流程，起始活动不是添加活动。

* 流转条件

表示从一个活动（源活动）转到另一个活动（目标活动）的条件，流转条件包含一个结果为真(true)或假(false)的逻辑表达式，当表达式为真时，目标活动将被启动，当表达式为假时，目标活动不被启动。目前表达式支持VB的语法。在表达式定义过程中，流程表单中的每一输入数据项都可以作为变量被引用。流转条件包含的属性还有：类型（单径流转、多径流转）、文字描述、源活动、目标活动、名称等。流转条件在流程图中表示为一个连接源活动方框和目标活动方框的单向连接线。

* 单径流转

为流转条件的一种，当目标活动的活动人不唯一时，即一个活动存在多个活动人时，单径流转的处理逻辑是当其中一个活动人完成了目标活动，则此目标活动即变成完成活动。无须再做。

* 多径流转

为流转条件的一种，当目标活动的活动人不唯一时，即一个活动存在多个活动人时，多径流转的处理逻辑是当其中一个活动人完成了目标活动，则此目标活动仍为待办活动。只有所有的活动人都完成了活动，该目标活动才变成完成活动。

* 流程图

业务流程的图型表示。流程图有两种形式，一种为模型流程图。表示流程的模型。另一种为实例流程图，不但表示流程的模型，同时还表示流程的实例。即流程实例的完成活动和待办活动。

* 流程实例

流程启动以后产生的运行实例，即一个运行的业务流程。系统会产生一个唯一标识ID来表示流程实例。

* 活动实例（任务实例）

在流程实例中启动了的活动运行的实例，系统会产生一个唯一的标识ID来表示活动的实例。

* 完成活动（完成任务）

当一个活动实例已经完成，为完成活动。

* 待办活动（待办任务）

当一个活动实例没有完成，为待办活动。

* 超时活动（超时任务）

当一个待办活动超过活动设定的时限为超时活动。

* 流程代理

流程代理是如果用户离开，可以将自己的任务代理给别人执行。执行时相当于本人执行。

## 参考资料

# 流程需求

## 功能结构

[这里用组织图的形式将该模块的功能全部列出来，可以从整体角度了解该模块的功能和从属结构]

## 流程的一般需求

### 支持多种流程类型

#### 一般流程（Generic Process）

通常意义上的流程，按照预先设计的任务和路由进行流转。

#### 自由流（Unrestricted Process）

即动态路由。

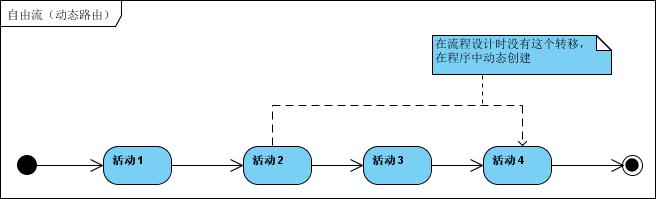
这种情况并不是太符合流程化工作，但是在OA 系统中应用很广，尤其是规范性要求不是很高的单位。

* 需求描述：

针对于特定的业务实例，在原本没有转移关系的环节之间进行特定的跳转。

在自由流中，有任务概念，可以根据所需执行的业务需要设置任务，但不再受路由的限制，可以从任何一个任务向任何其他的任务发送，相当于在所有的任务之间都有路由。

* 示例



### 多种路由方式

#### 串行路由（Sequence Routing）

即一个活动接着一个活动的基本路由方式。

#### 并行路由（Parallel Routing）

一个活动接下来分出多个同时可以执行的活动，通常会由同步网关分出。

#### 工作队列（Queues）【类似抢占式】

在企业内经常有“多人处理同一种工作”的情况。为提高工作效率，合理分配工作量，对于这种队列工作方式而言，合理的处理方法不是直接传送给特定个人，而是传送至Queue，Queue 的成员一旦有时间，便可向Queue要求接收新的工作。

说明：与抢占式处理有很大类似，但不完全相同。

举例：银行信用卡处理中心有3 位人员处理收件，10 位工作人员检查收入证明，3 位收件人员收件完毕后，将资料直接送给「收入检查」的Queue，则10 位收入检查人员的任何一位，均可向这个Queue 要求新的待审案件。

#### 发散路由（Emanative Routing）【多个实例】

即X3的任务分发。

企业内部有许多作业必需平行处理以提高效率，举例来说：有5 位部门经理需要提出年度预算报告，每一部门之报告为独立提出，故可将五位经理设置在同一任务内，各自处理后再统一送到下一任务。

一个实例经过发散路由时，将会根据需要复制为多个实例，各个实例将单独运转。

注意，这和并行路由有本质的不同。

并行路由会同时触发多个任务，以及生成各个任务相应的任务实例。

#### 聚合路由（Merge Routing）

针对上面的发散路由，对一个任务生成的多个任务实例需要在下面的某个任务节点进行汇总，汇总到一起后才继续往下处理。

#### 条件路由（Conditional Routing）

在企业处理日常工作时，有许多步骤只有在特定条件成立时才会执行。工作流程自动化软件因此必需提供此功能。

##### 条件表达式支持

即按表达式进行条件判断，执行满足表达式的分支。

对于分支网关、条件同步网关均需要设计条件表达式。

##### 根据表单中的选择路由

根据表单中的某个字段的选择来选择执行路径。

在X3中，是以表单某个字段作为流程变量，然后通过条件表达式处理变量值。

#### 关系路由（Relationship Based Routings）

大部分企业流程是构建在从属关系上的：申请差旅费需由部门经理核准、员工绩效由上级主管评定……等等。如果通过指定某人向某人汇报来实现关系路由显然不科学（对大的企业也不可能），所以能依据从属关系来决定流程传递方向的功能更显重要。

在X3中通过人员设置功能，根据人员之间的关系来确认某个任务的执行人。

#### 基于角色的路由(Role Based Routing)

角色是指权限分配的单位和载体。

基于角色的路由不同于以员工姓名为依据，如果职务发生变化（这在企业是屡见不鲜的常事），流程设计则不需变动。

#### 群体响应(即会签)(Group Response/Vote)

在企业运作环境中，有时需要对团体成员进行意见调查，如果全部或超过一定数目的人表示意见，则此程序或工作便继续往下进行。

### 路由功能

#### 办理（Complete）

任务的正常完成，完成后按流程定义跳转到下一步。

办理可以录入审批意见。

#### 在办

“在办”状态是为了方便管理人员对流程任务进行监控。

用户接到一个任务后，可以根据自己的工作安排，在真正要开始处理该任务时，可以将任务置为“在办”，办理需要一段时间，这段时间内可以通过写备注来注明工作进展的情况，以便管理人员查看。

#### 备注

用户可以自己的任务撰写备注，和审批意见不同，备注仅是一段说明文字和附件，不会影响流程流转。

#### 交办（Assign）

执行人因某些情况无法办理当前的任务，可以将该任务指派给其他的执行人。

#### 收回（Process Callback）

* 需求描述

在流程已到达下面的步骤的情况下，由前面已完成步骤的执行人收回此任务，即任务直接跳转回前面的的步骤。

* 示例

业务人员依照客户要求填写订单后，订单送出往下继续传递，隔了一天后，客户临时决定要更改订货的内容，您可以在不删除订单流程的情况下，使用反向回传的功能，可从有问题的步骤(订单输入)直接「取回」已流到后面数个步骤的该张订单，修改完毕后再送至下一步骤.

一般这种情况，实际系统实现中，会强制在后续处理人未处理的情况下可做出[取回]活动，否则不能取回。

#### 自循环（Self-Cycle）

在电子政务办公系统中，经常出现的“多处长联合审批”过程。多个处长（个人）属于同一个处长角色（角色单元）。针对同一个审批过程，采用自循环（审批这个过程重复执行）就可以基本解决问题。

和发散路由的串行模型不同的是，自循环是在当前任务实例完成后，下一步依然是指向当前任务，指导循环结束。

这样就不存在汇总，也不存在多任务实例的情况。

#### 回退（Process Returns）

也称驳回。

这项业务因为各种原因（文档不全、发送错误等等），当然处理人要求上一处理人重新办理，或重新发送。

#### 跨环节回退（Cross Returns）

* 需求描述

也称跨环节驳回。

不仅仅局限于退回上一步，而是可以根据流程设定，退回此业务流经的某一个（或某几个）任务。

* 示例说明

流程图：

****

例如，在第5个审批环节，审批不通过时，直接回退到第2或第1个环节。而到底回退到哪个环节是可以让用户根据业务需求进行自定义的，并且在回退环节工作完成之后，其下一步的方向也可以让用户自定义。如，要是由第5个环节回退到第2个环节，那么当第2个环节重新修改业务数据并办理完毕后，流程引擎可以设定是重新按照2-3-4-5的顺序重新执行一遍，也可以设定由第2个节点返回给第5个节点，由第5个节点重新审批。

#### 选择路径跳转

由执行人人工选择要走那条分支路径，而不是由表达式来自动判定。

#### 自由跳转

由执行人选择接下来跳转的目标任务，此功能一般需要慎重使用，否则会出现业务数据的不匹配。

#### 回执（Receipt）

执行人打开当前任务时，给到达该任务的前驱环节的执行人发送一条回执通知。

如在财务报销流程中，申请者想在第一时间里知道他的上级领导是否已经打开他的申请单。

#### 拒收（Rejection）

当前执行人因为某种原因拒绝接收任务，将已经发生的流转退回到前驱环节。

拒收和回退不同的地方在于：

前驱环节的执行人在接收到到退回的任务后，修改再办理完成时，不能再发给原先的执行人，只能选择其他的执行人进行发送。

#### 签收（Sign For）

执行人打开当前任务时，可以点击“签收”，系统会发送一条回执通知给前驱环节的执行人。

#### 协商和协商返回（Consult Returns）

居于回退和拒收开发的功能，增加了协商的确认环节。即当前执行人提出回退申请，如果前驱执行人同意，则可以完成回退或者拒收的操作。

#### 取消流程实例

业务管理员可以手工取消流程实例，让其直接结束。

#### 委托代理（Agency）

##### 自动委托/出差委托

* 需求描述

【代理未来工作(Assign Future Tasks)】

如果某员工即将请假或不在办公室，通常他会交待同仁哪些工作会在他不在的时间内发生，或预期可能发生。因此工作流程软件必需提供此项功能。

即A员工可以指定在未来某段时间内，委托B员工代理A参与的某个流程。

如果A员工参与到多个流程，则可以就每个流程分别委托代理。

【注意】选择委托员工时，系统必须判断该员工是否拥有该流程和该环节的相关授权。

* 需求细节
  + 委托协商

即选择代理人是否需要需要在线协商，还是线下完成？协商的内容包括受托人是否愿意接受委托以及该次委托的时限。

* + 委托时效

委托必须是有时间范围的，而且一般是短期，如果时间较长，建议直接修改流程定义中的人员设置。

* + 委托范围

所负责的全部或者部分流程。

* + 委托条件

委托双方必须具备的一些角色权限，如委托权限，办理指定流程、指定任务环节的权限等。

* + 委托撤销

委托关系可以中途撤销。

* + 委托跟踪

关于受托人对委托内容的执行情况的跟踪和记录，可以查看历史，防止恶意行为。

* 其它
  + 关于委托的粒度

目前是基于流程级别的，即如果A执行人将流程委托给B，则A在该流程中的负责的所有任务均委托给B。

是否有必要将委托粒度设置到任务级别呢？

##### 手工委托

【由用户指定代理人(Client Assign Functions)】

执行人A若因有其它事情导致无法处理工作时，可以指定B员工为其当前流程实例的代理。

即该流程实例内，所有A负责的任务均有B代办，当该流程实例结束后，委托关系也结束。

【注】可以选择代理人应该是在某个指定范围内的，而不是全系统的所有员工，这和权限、委托设置等有关系。

##### 委托组和动态指派

上面阐述的是指定特定的人员为代理人，考虑到灵活性，可以考虑支持委托组。然后根据优先级或者工作量百分率等算法动态指派，这样可以有效的将工作量分散给多个代理人。

#### 补发功能

针对发散路由（分发）设计的功能，分发是一次性产生若干个任务实例，但如果在执行过程中，发现漏了某个用户，或者新加入了某个用户，那么可以就该用户补发一个任务实例。

如采购询价，选择了若干供应商进行报价，报价过程中，加入了新的供应商，可以对这些供应商补发报价任务。

#### 工作重送(Resend Steps)

当原始工作程序有问题或设备故障导致流程错误时，流程管理员可将工作重新发送，以避免流程无法继续进行。

#### 重新指派执行人

【能临时更改收件人(Ability to Change Recipient )】

我们通常会需要介入某个正在进行中的任务，并将某个步骤收件人临时更改为指定人员处理，此功能对于例外处理或未预期的特殊状况非常重要。

此功能和2.2.3.11.2手工委托原理一样，但通常由流程的业务管理员来使用。

用于解决按照正常流程无法继续办理下去的业务，如：实例的当前办理人出差而且没有设置委托，特殊实例需要跳过若干环节。

而且由于业务管理员拥有更高的权限，所以可选的新执行人的范围会比较大，甚至是系统全部人员。

#### 备注

任务除了可以直接办理外，

### 功能的条件化设置

#### 条件化步骤（Conditional Steps）

在企业处理日常工作时，有许多步骤只有在特定条件成立时才会执行。工作流程自动化软件因此必需提供此功能。

通过设置条件表达式，针对分支、同步、条件同步各类网关。

#### 条件化步骤跳跃（Conditional Jumps）

条件式跳跃指满足某些特定条件时，必须自动跳过中间数个步骤至指定人员处理。这也是企业工作程序里屡见不鲜的状况。

##### 条件化取消流程（Conditional Aborts）

在企业内常发生当遇到某些状况时，则整个流程实例便取消而不再流转。工作流程自动化软件也必需相对提供这项功能。

##### 条件化退回（Conditional Returns）

业务流程可能因资料错误或上级不同意其内容而将资料退回上一步骤（也可能退回到流程发起人）。工作流系统必需支持此功能。

##### 条件化执行人（Conditional Recipients）

在许多企业环境里，工作的分派是依照各人的职责或它的专长。因此，工作流程自动化软件必需提供依实际状况决定分派工作给谁的功能。

##### 条件定义清单(Event Condition Tables)

现代企业组织内，每天都要面对各种例外状况与特殊事件。因此，逻辑判断与例外处理功能是否强大，是决定工作流程软件优劣的重要指标，它可以依照企业内的规范，以及个案的特殊状况，聪明地将工作传递到正确的处理人员手上。

###### 条件定义清单与其它步骤互动

在许多情况下，我们必需由其它步骤的处理状况(或现况)来决定工作/决策的未来动向。应该要求软件实现提供其它步骤目前的状况信息，工作流设计工具能使用其它步骤的现况资料，来定义此步骤条件清单以控制资料的流向。

#### 条件化执行人（Conditional Recipients）

前面一节对此功能做了描述，在X3中，表现在流程设置中的人员设置功能中。

X3是表单规则+用户选择规则 两者合成一条人员规则。

即在满足该表单规则的前提下，按照该规则的用户设置查询执行人集合。

X4中将会更改为流程变量规则 + 用户选择规则。并对原有的代码按策略设计模式进行重构。

目前支持的用户选择规则包括：

1. 发起人
2. 用户
3. 角色
4. 组织
5. 组织负责人
6. 岗位
7. 上下级
8. 用户属性
9. 组织属性
10. 与发起人相同部门
11. 与其他节点相同执行人
12. 发起人的直属领导（组织）
13. 脚本
14. 上个任务执行人的直属领导（组织）
15. 发起人的领导
16. 上个任务执行人的领导
17. 部门的上级类型部门的负责人

### 活动属性

#### 流程实例暂停(Delayed Assignment)

在许多真实的情况下，我们通常会暂停某特定工作一段时间，以便等待其它事件或工作完成。

即允许业务管理员对流程实例进行暂停，该实例正执行中的任务实例也同样被置暂停状态，并冗余记录暂停前的状态以便恢复。

#### 禁止代理的工作(Non-assignable User Tasks)

员工常会请别人代理自己的工作。但是，有某些工作必需由员工本人亲自处理，而不能请人代理的(例如某些重要的报表填写、牵涉公司重要机密的信息、个人薪资文件…..等)，因此工作流程系统必需允许设计者定义哪些工作是可代理，哪些是不可代理的。

### 时限需求

#### 流程时限

整个流程运转的时限要求，可能是有3中类型：

* 没有期限（默认）
* 有期限
  + 相对流程开始 或 相对节点创建或完成 的时间期限；
  + 相对时间可以是工作日 或 自然日。

#### 时限类型

流程中某一活动的时限要求，可能有提醒时限、警告时限、超时时限。

根据时限类型，当到达设定的时限时，会触发相应的事件，如发送邮件、站内消息、短信通知、生成催办、通知部门主管领导、通知监察部门和执行指定的动作（如直接跳过这一步骤、指派给其他人、结束流程等）。

#### 时间类型

分为工作日和自然日两种。

工作日需要通过系统的工作日历进行维护。

自然日无需维护。

##### 工作日历

每一家企业针对国定假日、例假日等或多或少都有自己的计划，在计算流程工作日与逾时等统计上，必需要把假日扣除，因此工作流程自动化软件必需要提供假日定义的功能，以免在计算上述时间发生错误。

#### 停止和继续计时

对于审批类业务，在审批时限内需要完成审批，而在专家论证、现场考察、检验检疫等环节，如果时间不固定，需要停止计时（如整个流程涉及绩效考虑等）。

#### 事件通知

短信、邮件、站内信息、客户端信息等。

#### 自动催办

针对警告时限，可以设置催办，催办除了发送催办信息外，另外可以有针对性的设置任务状态或记录一些信息供绩效考核使用。

#### 到期动作处理

在某个处理点上停留时间超过标准处理时间，可以执行到期动作，启动某处理步骤，或将任务转交给他人。

目前X3支持的到期动作有：

1. 无动作；
2. 执行同意操作
3. 执行反对操作
4. 执行驳回操作
5. 执行驳回到发起人操作
6. 执行交办操作（需选择指定的交办人）
7. 结束该流程
8. 调用指定方法（即设置一段脚本供执行）

### 数据处理功能

#### 以流程变量存储表单数据

通过维护变量和表单的映射关系，将表单和流程变量进行捆绑。

显示表单时，通过流程变量的数据进行填充。

保存表单时，将数据更新回流程变量。

流程变量持久化到数据库，方便在流程实例和流程历史中查看。

#### 利用变量传递数据

通过将流程节点和变量进行关联，可以让变量在流程中各个节点中存在和流转，然后通过表单显示和修改、通过脚本来进行操作。

#### 业务规则数据捆绑变量

业务规则数据可以：

1、启动流程时，从外部系统传递进来；

2、在编写表单数据，更新到流程变量中，即以表单某些数据的选择作为业务规则数据。

3、系统其它规则生成的数据。

这些业务规则数据一概保存到流程变量中，在流程定义中，可以通过脚本来操作这些数据，以完成该流程的相关业务处理。

#### 流水号支持(Serial Number)

有时需要从特定数值开始起算项目代号或案件代号。(例如

20130528-00001)，在命名流程实例时，通过系统的流水号提供支持。

## 流程的高级需求

### 流程定义版本控制(Definition Version Control)

因为业务流程可能经常需要修订，能够易于安装与管理新版本流程也是极重要的必备功能。在修改流程定义时，根据业务需求，可以决定是产生新的版本，还是直接修改已有的版本。

### 流程定义的状态控制

一个流程可以有多个版本，这些版本可以同时运行在工作流引擎中，而针对每一个流程的版本，都可以有多种状态。

目前设计了未发布、发布和禁用三种流程。

### 流程的分级管理

#### 分布式用户管理(Distributed User Administration)

在大型企业里，集中管理所有用户工作往往不切实际，用户管理工作必需分散，使得部门主管能直接管理其部属。

即允许子组织的主负责人或者指定的用户管理员（HR）自行维护该组织的自组织和组织用户。

#### 流程的独立管理和授权

同流程授权。

通过系统角色指派设计流程的权限，流程创建者默认是该流程的管理员，拥有流程授权。授权的类型有修改设计、修改设置（可以更细化到每个设置点）、查看和参与流程等。

目前X3已经实现了一定的授权功能，需要在此基础上做进一步的扩展。

### 子流程需求

#### 外部子流程

工作流自动化解决方案，必须能够由主流程呼叫子流程，从而实现网状工作流程的设计。例如请购流程内的订单核准步骤通过后，便自动启动验货流程。现代企业工作往往非常复杂，这项功能可保证流程设计与导入能够更为模块化。

#### 内嵌子流程

在流程的设计过程中，常会发现有不少相同的步骤，如：国家发改委的建议提案项目中，文件在各司局内部办理时，其过程完全相同，不论是主办、会办还是会签流程，可以将此部分作为一段子流程，在多个流程中使用，使得流程的设计更加简便、快捷、准确，今后这部分业务发生变化，进行维护时也只需修改此子流程。

和外部子流程不同的之处在于，内嵌的子流程虽然是独立的，可以为多个流程所共用，但是它不可以独立启动。

#### 内部子流程

即在流程设计时，将若干步骤归入一个内部子流程中，简化流程的设计和管理。并且可以通过复制的方式，将该子流程复制到其它流程中。

#### 子流程支持同步和异步

当主流程呼叫子流程时，需要提供以下的弹性选项：主流程必需等待子流程结束后再继续进行(同步启动)；或主流程直接进行下一步，而毋需等待子流程结束(异步启动)。

#### 主子流程数据交互

当主要工作流程呼叫另一支子流程时，必需要能够传送数据到子流程内；当子流程结束后，它也必需能将数据传回主流程。

系统需要能够支持前面三种子流程类型的主、子数据传入和传回。

#### 可传回子流程统计数据

当企业流程牵涉了一个或多个子流程时，计算该企业流程的整体成本与费时统计，便需要包含子流程的相关数据。

#### 子流程的循环、递归调用

在主流程中，可以多次反复调用子流程。而在子流程中可以调用其他的子流程，甚至调用自己，形成递归调用。

### 动态路由(Dynamic Routing)

能够依照实时信息（条件表达式或脚本），动态变更流程跳转逻辑，允许在两个没有直接连线（转移）的任务进行跳转，但不会修改流程设计。

## 流程设计器

### 图形化工作流程图

以直观人性化的图形化方式设计企业工作流程图，已成为现今工作流自动化软件必备的基本功能。每一张流程图代表了一个业务流程，同时可定义个别步骤的意义、条件判断与工作处理顺序和路由等，便于流程管理员设计各种企业流程。

### 连线和转折点

两个节点（事件、任务和网关）之间的连接线，并且自动生成转折点，让线条更好看。

### 事件节点

X3提供了开始、结束和异常结束三种事件节点。

在实际业务中，开始和结束节点均有多种类型，但为了简化设计工具的复杂度，我们将更复杂的设置放在流程设置中。

如对于开始节点，还有其它的诸如条件开始、消息开始、定时开始等类型。

对于结束节点，目前允许设置一段脚本用于补偿处理，但在以后可以增加一些常用的功能或脚本，简化补偿脚本的维护难度。

### 任务

目前支持如下的类型。

* 用户任务
* 脚本任务
* 消息任务
* WebService任务
* 会签任务
* 外部子流程
* 内嵌子流程
* 泳道（水平、垂直）

### 网关

* 分支
* 同步
* 条件同步

用不同的图形表示各个网关，相应的在流程设置中，对网关的表达式设置也有所不同。

### 复制流程步骤

流程里的某些工作，往往本质十分相近。(例如某些单据副理签核完毕送交经理签核，两个步骤的表格与逻辑几乎雷同，只是签核人不同而已) 因此在设计流程时，如果软件能提供复制功能，将步骤或整个流程进行复制，再修改差异部份，将使得设计师不需要重复设计相同的电子表格或定义逻辑，大量节省流程设计时间。

### 流程自动排版

提供各种自动排版功能。

### 流程设计文件

将企业工作程序制成文件是非常重要且必须的，在所有的项目中，用户及监理方都会要求将所有流程形成标准的文档，并且，这也是在流程定义完成后，检查流程定义是否符合业务需求的重要手段，因此，工作流自动化软件因此必须提供自动生成流程过程文件的功能。

流程设计器允许将流程保存为idfl文件，也支持打开该类文件。这样方便在多个系统中传输。

### 群体开发流程

正如无法期望由一位设计人员开发整个大型流程，很多时候工作流系统必需支持整个团队同时参与的群体开发架构。

涉及多个流程设计人员时，需要提供设计锁定功能，即任何时候只有一个人员能够编辑和保存该流程。

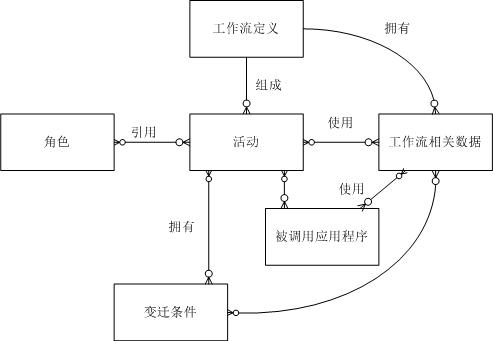
### 流程定义的自动检查

在流程定义完成后，或者流程定义切换为发布状态之前，应该自动的检查流程定义是否有逻辑上的错误，并指出出错的位置，以便及时改正，避免对流程进行无意义的调试。

## 流程管理

### 流程定义管理

下图是流程定义的元模型，流程定义管理相关功能就是基于这个元模型来规划的。



#### 流程列表

分页方式显示流程，根据流程状态显示相应的流程管理功能的入口。

#### 流程列表的管理功能

* 在线流程设计：新建流程定义
* 导出：将流程定义和相关的设置导出为xml
* 导入：将导出的xml导入到系统内，一般用于从开发系统导入到正式运行系统中。
* 删除
* 设计：进入流程设计器对流程设计进行编辑
* 设置：集中了流程定义大部分功能
* 启动/发布/禁用：根据流程状态显示。
* 授权：允许相应的用户看到此流程。可以进一步增加限制用户参与流程。

#### 流程设置

详见X3的功能，在X4中，将并根据前面的流程一般需求、流程高级需求说明的需求增加相应的功能点在这个页面中。

### 流程仿真测试

业务流程牵涉了大量用户与设备环境，把新完成的工作流系统安装到现实环境中进行测试，是非常不切实际和低效率的，毕竟企业不是“小白鼠”。如果工作流解决方案缺乏这一功能，在实施过程中将给企业造成损失。流程仿真功能可以让设计者在模拟环境中直接仿真测试。

仿真测试涉及几个方面：

1. 模拟环境的搭建

模拟环境由IT人员搭建，录有若干实际的业务数据（去掉保密数据）

1. 仿真测试子系统

通过仿真测试登录入口，可以登录到仿真测试子系统，该界面只显示流程仿真测试相关的菜单、栏目和功能，提高测试效率。

1. 用户快速切换

用户登录后点击“开始仿真测试”，则会弹出一个“用户列表页面”，显示该流程涉及的所有用户，该界面提供快捷登录方式，每个用户旁边均有“登录”链接，点击后直接修改Session，无需经过登录页面。

1. 测试报告

记录从点击“开始仿真测试”到点击“结束仿真测试”之间的所有操作日志，并做分析，生成测试报告。

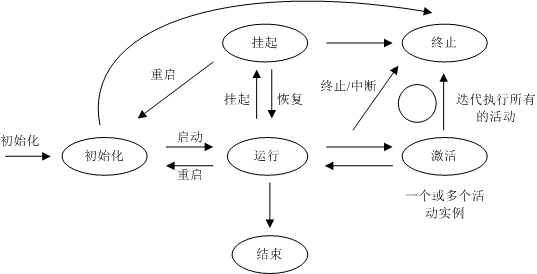
1. 数据回滚

删除上面开始到结束期间生成的数据，保留干净的模拟环境供下一次测试使用。

### 流程实例管理

在工作流执行服务中，流程实例能够响应外部事件并改变自身的状态，比如执行一项任务、激活一个活动或者响应由工作流引擎指定的控制逻辑，又如调度流程实例的下个活动或任务。因此，工作流执行服务可以看作是一个状态变化机，用来控制流程实例和活动实例的状态转换。

能够支持实例的多种状态，按照WFMC的模型，其实例有6种状态：Initiated / Running / Complete/ Active/ Terminated/ Suspended。状态图如下：



在X4系统中，将提供管理界面对流程实例各个状态进行维护，并相应的提供该状态下的功能。

### 流程任务管理

管理流程任务实例。

* 任务查询

提供多种任务查询方式，主要根据权限级别进行区分。

* + 管理员：查看所有任务；
  + 组织负责人：查看该部门所有待办任务。（私密任务除外）
  + 人员：查看自己的任务、代理的任务。
  + 领导：查看直接下属的任务（私密任务除外）。
* 任务操作
  + 待办：接收到发给自己的任务后，未曾办理。
  + 已阅：对任务第一次点击进入后，标记为已阅，并同时会发一份回执给相应的信息接收方（如任务发起人、上一个任务的执行人、业务管理员等等）
  + 办理中：提供这样的状态对需要一定时间办理的任务有用，即用户根据自身工作任务排期，当前任务正在办理中。办理中的任务可以写备注。
  + 完成：点击完成，任务移交到下一步。

### 流程操作管理

流程任务实例在完成后会生成一条流程操作，查看操作历史，可以了解整个流程实例的处理过程。

### 流程历史管理

流程实例完成后，该条实例会转到流程历史表中存储，避免流程实例表过于庞大。

可以查看流程历史的明细和操作日志。

## 流程监控管理

### 监控所有实例

和流程管理功能中的查看所有流程实例和任务实例不同的地方在于这里更着重对流程的运作情况的监控。如正常运作的，超时，停顿，发生异常等等。

一般采用图形加列表的显示方式。

图形化用于监控所有状态，显示时间分布和统计数据。点击后进入列表或者实例详细页面。

### 监控组织内的所有实例

同上，只是监控的实例数据是该组织内的实例。

### 可同时监控主/子流程

当主流程激活一个或多个子流程时，能否在监控主流程时同步显示子流程现况，对管理者与用户都是一项很实用的功能。

关键是在监控页面中，连子流程的运行状态也显示在主流程中。

### 图形化工作流程监控

工作流自动化最重要的效益之一，就是可以图形画面显示任何流程的处理进度，以取代人工操作中传统且无效率的的以人力追踪管制。

本模块关键难点将监控结果用图形化展示，并自动刷新。（如每隔1分钟刷新一次，如数据量太大，则需要将间隔设置长一些）

### 异常状况通知

当企业流程工作中发生异常问题时，必需能马上通知流程处理人员，并提供错误发生原因等必要信息。缺乏异常通知的系统，会产生决策误导、混淆认知与提供错误信息等问题，因此工作流程自动化系统必需提供异常通知功能，以协助尽快了解并处理问题。

信息通知和2.2.6的时限需求有些交叉点，设计时可以考虑如何重用。

### 处理停顿流程

当执行工作流程时，有可能某一流程会“卡住”而无法继续往下执行，此情况通常是发生在不兼容的条件状况，而无法使流程进行到正确的后续步骤(例如无限制的重复代理)，因此工作流解决方案必需提供侦测与回复异常停顿流程的功能。

允许监控人员（需拥有相应的权限），或者联系相应的流程管理员对这些流程实例进行操作，如执行完成、回退、结束流程等。

# 表单模型、流程变量和表单

## 表单模型

在系统设计表单模型数据结构，用于实现复合数据类型，框架将表单模型生成动态类，并和物理表做ORM映射，这样在保存表单模型时就直接保存到数据库中了。

流程变量可以选择对应的数据类型，除了Java基本类型外，还可以选择表单模型（复合数据类型）。

表单模型从属于流程定义。

表单模型用于流程变量和表单建立映射关系时，提供复合数据类型结构的支持。

### 表单模型权限

在表单模型设置的权限，是表单中的默认权限。

通过表单模型 <--> 流程变量 <--> 表单 的关联关系进行获取。

## 表单类型

### 内部表单

创建内部表单时，可以将1～多个流程变量和表单元素进行关联。

1、如果流程变量是基础类型，则表单元素关联的是流程变量的本身。

2、如果流程变量是基于表单模型创建的，则表单元素关联的是流程变量的某个成员变量。

### 外部表单

#### 第三方表单

在系统内的流程中使用外部第三方系统的表单，并且表单的数据和流程的流转密切相关。

这时候便需要通过流程变量和第三方表单进行映射，在建立映射后，表单的填充和数据保存将会通过流程变量完成。

这样在流程设置中，便可以根据变量的值设置相关的条件表达式和脚本。

#### URL表单

系统独立开发的业务表单或外部系统的表单，在流程的表单设置中，通过URL方式配置，通过前置或者后置处理器完成表单数据的读取和存储。

## 流程变量

通过流程变量维护流程的所有数据，包括业务数据和流程数据，根据流程变量

## 变量表单映射

1、对于内部表单：由于在建表单时，根据流程变量（变量或者变量成员）和表单元素的命名关联，直接生成变量表单映射。（该映射集合用于表单定制权限的维护）

2、对于外部表单：需要通过开发分析外部表单，或者人工将外部表单的数据维护到系统内，然后将变量和表单进行映射。

## 表单的展现

表单的数据从流程变量中来，根据流程变量和表单的映射关联，用流程变量填充表单，然后展现表单。

## 表单的权限

除了使用表单模型默认的权限外，在维护流程变量时，可以设置定制的权限，并且将这个权限集关联到该流程定义内1～多个任务中。

## 表单的数据验证(Data Validation)

在设计表单模型时，对各个字段的数据验证方式，在设计表单时，将字段（通过流程变量关联）和表单元素进行关联，自动在页面上生成该元素的验证表达式。

在编辑表单数据时，通过JS方法对数据进行验证。

## 表单数据格式化输入(Data Masking)

数据的填写和输入必须有一定的格式。(例如日期为mm/dd/yy)，工作流自动化系统当然必需提供此功能，以确保数据填写的正确性与方便性。

### 自动填写

* 新建表单时，根据脚本计算，自动填入数据。
* 第一次保存表单时，根据表单数据，通过脚本计算，自动填入数据。
* 执行到某一步骤时，根据脚本计算，自动填写。

## 表单打印

设计打印模板，允许对表单部分的内容进行打印。

## 表单缓存(Forms Caching)

电子表单是业务流程的用户界面，因此可能被重复使用很多次，所以，若能对表单利用缓存方式(Caching)储存于客户端，将提高速度与可用性（因为表单可能包括图像或其它类型文件）。

组装表单是比较重量级的操作，如果将流程变量，变量表单映射、表单等数据做缓存处理，这样在处理任务时，显示表单的速度将大幅提升。

## 表单设计器

在项目中通常会有大量的表单，常见的办公业务中通常有十几个到几十个表单，而审批系统中常常有数百甚至上千个审批事项，这使得表单的快速搭建非常重要。

提供可视化的表单设计器将大大降低设计表单的技术要求，通过集成的快速设计功能，可以在零编码的情况下完成表单的设计和配置。

## 表单数据模版 (Form Data Templates)

某些特定的流程工作是在不同的时间执行相同的内容，此类工作最好能有一份“标准范例”，范例内包括所有标准内容，用户下次执行此工作时，毋需再重复输入相同的内容。

这里指的是内容的模板，而不是设计的模板。

即每次启动流程时，可以基于已填充好的内容模板，这样便可以大幅降低每次填充相同数据的工作量，以及降低出错的几率。

## 表单数据自动归档(Automatic Archiving)

由于每次启动流程都会在表单对应的物理表（1～N张）增加至少一条记录，当系统使用时间越长时，相应的数据量也会很大，但每次启动新流程实例时，依然是访问这些表，这样将会导致查询数据减慢。

可以考虑在生成物理表时，同时生成对应的归档表（结构一样，命名不同），在流程实例结束后，如果流程实例的处理一样，将表单的数据归档到表单数据归档表中。

# 组织模型和权限模型

## 组织模型的抽象

* 常用的实体

组织架构中明显的业务实体有三个：

机构——为完成某种职能所建立的实体单位。机构之间通过上下级关系形成树形结构。

岗位——在各级机构中设立的专门用于处理某方面事务的职位。

员工——企业聘用来承担某种具体工作的个人。

* 该模型表达了以下的业务逻辑：
  + 机构是层次性的，机构之间通过上下级关系形成一棵单根的树。除总部之外，每个机构有且只有一个上级机构。
  + 每个机构下面可设立多个岗位，每个岗位只属于一个机构。
  + 类似于机构，岗位之间也通过上下级关系形成树形结构。除最高岗位之外，每个岗位有且只有一个上级岗位。
  + 岗位与员工之间是多对多的关系。每个员工可担任一或多个岗位，每个岗位可容纳0或多个员工。
  + 员工与机构之间没有直接的隶属关系。员工通过担任某机构下的岗位间接地与机构之间形成隶属关系。
* 模型演化和抽象

通过上面的模型演化，隐约发现深藏的领域抽象——当事人和责任。

以“当事人（Party）”概括表示机构、岗位和员工，以“当事人关系”表示 两个当事人之间在一定的时间范围内存在的各种。

最终得出组织架构模型的本质：各种类型的当事人以及它们之间的责任关系。

## 组织功能

### 企业整体组织管理

提供对整个企业组织结构、层级从属、主负责人、人员关联等管理的功能。

### 子组织分级管理

在大型企业或单位内，很少有人能认识全体员工、并了解他们的职称、工作内容、直属主管以及任何职务异动。因此最好办法，便是让许多不同的人管理自己的企业组织图，各自维护不同企业组织图的最新异动状况。

可以由各个子组织的主负责人或者指定的人员（如HR）进行该子组织的维护。

### 跨部门群组

某些工作是由一群特定人员或者小组处理，甚至必需跨部门组建群组。所以系统需要支持定义群组，然后从组织各个部门中选择成员加入该群组，群组一般只属于根组织。

### 动态定义群组

“群组”（或我们熟知的“项目小组”）常常是为了完成特定工作而成立的编组，而工作流软件必须能定义并使用动态编组功能以适应这种业务需求。

所谓“动态”是指能在流程执行时动态指定群组成员，而非在流程设计时。

在某个流程中，不同的流程实例，某个步骤指派的人员集合是有可能不同的，而且该动态群组形成后，在该流程实例内会反复用到，但在该流程实例结束后，该动态群组也同时解散。

### 公用用户

许多的流程是由大量组织内或组织外的不确定人员发起的。例如政府机关，允许全体市民提出建议或服务需求，再由机关内的人员加以处理。

如果这些流程需要这些组织外人员参与和操作，那么就需要制定一些公用角色和相应的公用用户了。

### 循序群组（自动委托组）

常需要特定群组中任何一人批准文件或工作(例如，50 万元采购要由副总批准，但副总不在请总经理或执行副总代批也可（这些人都属于公司“高级主管”群组中的一员）。

对应系统功能就是当副总收到此任务时，可以点击“自动委托”，则任务按照该群组的设置自动委托给下一个组内用户，而不需要副总直接指定。

### 权重群组

用于将任务按权重动态分配。

例如某一部门内张三﹑李四﹑王五的工作量分别设定为10%﹑20%与70%，则指定到此部门的工作流程将有10%被机动分派给张三﹑20%分派给李四，70%分派给王五，只要修改权重定义便能更改每个人的工作负荷量。

## 人员选择器

### X3已支持

1. 发起人
2. 用户
3. 角色
4. 组织
5. 组织负责人
6. 岗位
7. 上下级
8. 用户属性
9. 组织属性
10. 与发起人相同部门
11. 与其他节点相同执行人
12. 发起人的直属领导（组织）
13. 脚本
14. 上个任务执行人的直属领导（组织）
15. 发起人的领导
16. 上个任务执行人的领导
17. 部门的上级类型部门的负责人

### 考虑的扩展

1. 指定群组（自动委托）：对应4.2.6
2. 更多类型或者细化已有的的支持，以便让组合更加灵活。

## 权限模型

### 基于核心RBAC模型设计权限

关于核心RBAC模型网上有很多相关资料，这里不做冗述。

对于系统而言，角色（Role）将会直接和组织模型中提到的当事人（Party）进行关联，这样便给组织、员工或者其它当事人类型分配角色。

### 流程权限

#### 启动新流程的权限

在企业内部，有些流程发起的权限会被限定与管制，并非每一个人都可以自由启始任何流程（例如：技术部门工程师不可能去启始新的业务预估报表流程）。因此，系统需要提供控制哪些人可启始新流程的权限管理。

#### 可定义流程管理员

企业流程中往往会有某人负责某项工作的品质与进度，我们称之为“流程管理者”，例如项目经理、业务主管、会计组长、工程部经理….等业务执行与管理人。这些人必需确实掌握与工作相关的流程最新进度，从而调整流程、临时分派工作….等。因此，系统必需提供可指定流程管理者的功能。

除了流程管理者之外，实际业务中有更丰富的业务需求，如：业务管理者、流程读者，内容读者，安全审核员等等，应该提高更丰富、更灵活的运行和管理机制。【流程授权的扩展】

#### 任务权限

即任务的人员设置，指定哪些人员可以参与任务或者接收通知。

#### 表单权限控制

见3.6表单权限的说明。

# 个人中心

## 任务清单

## 列表的需求

### 工作列表显示（待办、在办、代理）

显示各类任务的列表、分页，并根据业务类型显示相应的功能按钮。

### 已处理工作列表（审批、发起）

即显示已经处理的任务列表。

### 按日历显示任务

按工作日历显示任务，有助于员工对近期要办理的所有任务有总体的了解。

允许员工对每个任务设置计划办理时间，在日历中按这个时间将任务显示各个日期栏中。这个计划办理时间有助于员工自行安排每周、每天的工作内容。

### 依重要程度优先级分组显示和排序

提供优先级设置，并在列表排序中按重要程度优先级分组和排序。

优先级默认值由流程设置时设置，但员工可以手工变更任务的优先级。

### 依要求完成时间缓急分组显示和排序

根据任务的时限设置，依据任务要求的完成时间进行排序，时间越前，排在最上面。

可以提供一些快捷过滤方式，比如：3天内必须完成的任务，一周内必须完成的任务。

## 显示进度落后的工作

列出收到过提醒通知、警告（催办）通知、以及超时但没有更改状态的任务。

## 用户监看流程实例

用户可以查看自己参与的所有流程实例的处理现状。

## 任务提醒

当有新的任务待处理时候，通知处理人。任务提醒的方式有很多中，比较流行的方式有三种：邮件,客户端任务提醒软件，短信通知。

短信通知

客户端提醒软件

WEB自动弹出信息；

IM弹出信息。

## 个性化

### 用户自订工作窗口画面(Custom Client Views)

用户往往希望用他们最习惯或最喜欢的方式看到有哪些待办事项，因此，工作流软件客户端必须提供一定程度的自订功能，让用户自行依他所喜好的风格呈现工作清单。

## 部门新成员的设置初始化

将一个新成员加入到某个部门后，可以通过初始化操作将一些部门相关的权限、流程授权、文档交付、入职考核等等工作一并完成。

对于某些功能可能是通过继承的方式获取，但某一些则是直接和用户关联，这些就需要在初始化中实现了。

# 统计

## 按照部门、类型、时间段进行统计

## 绩效统计

## 工作完成时间(Complete Time)

在业务进度控制中，通常会指定该工作所需的处理总时间(工时：例如10 人天，20 人月)，工作流程每一案件的处理总时数，是统计每一步骤实际处理时间而成。当企业工作流程自动化之后，这项工作就可以由自动化软件代劳。

## 工作负荷量显示(Workload View)

工作流自动化可以集中管理/显示所有员工目前的工作负荷，更重要的，便是能将指定工作或任务临时调派给其它人员处理。

## 工时成本(Task Rates)

对企业而言，每个参与工作流程的员工都是有成本的。这些费用应该被计算至项目成本或部门成本。工作流程软件必需提供成本计算功能，针对处理人员薪资不同而能统计流程处理成本。

## 任务成本统计(Task Cost Statistics)

工作流自动化软件必须能够产生流程中每一步骤的成本统计报表，以供管理人员了解人力成本结构并予以改善。

## 工作流程成本统计(Process Cost Statistics)

工作流自动化软件必需能够产生流程的成本统计报表，以供管理人员评估此流程的成本与效益。

## 单一步骤超时统计(Step Lag Time)

企业流程工作最主要的迟滞发生在传递时间(transfer time)与超时，亦即工作摆在待处理文件夹内尚未处理的等待时间，因此，必需要将此时间评估与统计出来。

## 工作流程超时统计(Process Lag Time)

特定项目的整体超时时间(所有步骤的超时时间)也是企业评估工作效率的重要依据，因此工作流系统必需提供统计此时间的功能。

## 工作流程信息输出(Workflow Metrics Export)

为方便使用者作更详细的分析，或产生各类统计报表，工作流软件应该支持将流程中产生的数据输出。用户可通过各类工具作进一步统计与再利用。

## 可自行定义统计报表(Configurable Reports)

工作流自动化软件必须能产生各类报表，以便用户了解流程的瓶颈与成本，对BPR 的工作提供足够的数据与参考。因为不同的企业有各自不同的需求，因此这类报表必需能让用户按需求自行定义。