



翻译人: 张敬波 OICQ: 42508298

Email: [say4ever2u@yahoo.com.cn](mailto:say4ever2u@yahoo.com.cn) MSN: [tabingfly@hotmail.com](mailto:tabingfly@hotmail.com)

翻译人: 韩伟 OICQ: 3413384

Email: [java\\_cn@21cn.com](mailto:java_cn@21cn.com) MSN: [Java\\_cn@21cn.com](mailto:Java_cn@21cn.com)

工 作 流 管 理 联 盟 规 范

# 工 作 流 管 理 联 盟 工 作 流 标 准

工作流过程定义接口——XML Process Definition Language

文档编号 WfM-C-1025

文档版本——草稿 1.0 (测试版本)

2002年7月31日

版本 1.0 (测试)

版权© 2002 WfM-C

2.	对象 .....	5
3.	目的 .....	5
4.	简介 .....	5
4.1.	一致性 .....	6
5.	过程定义内部转换总览 .....	6
5.1.	过程定义相互转换方法 .....	7
6.	元模型 .....	7
6.1.	实体总揽 .....	8
6.1.1.	工作流过程定义 .....	8
6.1.2.	工作流过程活动 .....	8
6.1.3.	转移信息 .....	9
6.1.4.	工作流参与者 声明 .....	9
6.1.5.	资源库 .....	9
6.1.6.	工作流应用程序声明 .....	10
6.1.7.	工作流相关数据 .....	10
6.1.8.	系统和运行环境数据 .....	10
6.1.9.	数据类型和表达式 .....	10
6.2.	过程、包 .....	11
6.3.	过程元模型 .....	12
6.4.	包元模型 .....	13
6.4.1.	过程库 .....	13
6.4.1.1.	重定义和范围 .....	14
6.5.	元素总揽 .....	14
6.5.1.	开发者扩展元素 .....	16
6.5.1.1.	扩展属性 .....	16
6.5.1.2.	扩展参数映射 .....	16
7.	XML 过程定义语言 .....	17
7.1.	多实体通用元素 .....	17
7.1.1.	扩展属性 .....	17
7.1.2.	形式参数 .....	18
7.1.2.1.	参数传递语义 .....	19

---

7.1.2.2.	并发语义 .....	19
7.1.2.3.	形式——实际参数 映射 .....	20
7.1.3.	外部参考 .....	20
7.1.3.1.	Web Services .....	21
7.2.	包定义 .....	22
7.2.1.	包定义头 (Package define header) .....	23
7.2.2.	Redefinable Header .....	24
7.2.3.	一致类声明 (Conformance Class Declaration) .....	26
7.2.4.	Script .....	27
7.2.5.	外部包参考 (External Package Reference) .....	28
7.3.	工作流应用程序声明 (Workflow Application Declaration) .....	28
7.3.1.	调用参数 (Invocation Parameters) .....	30
7.4.	工作流过程定义 (Workflow Process Define) .....	30
7.4.1.	工作流过程定义头 (Workflow Process Define Header) .....	32
7.4.2.	工作流过程可重定义头 (Workflow Process Redefinable Header) .....	34
7.4.3.	活动集 (Activity Set) .....	36
7.5.	工作流过程活动 (Workflow Process Activity) .....	37
7.5.1.	路径活动 (Route Activity) .....	41
7.5.2.	块活动 (Block Activity) .....	41
7.5.3.	运行控制属性 (Execution Control Attributes) .....	42
7.5.4.	活动替代执行方式 (Implementation Alternatives) .....	43
7.5.4.1.	No Implementation .....	44
7.5.4.2.	Tool .....	44
7.5.4.3.	subflow .....	45
7.5.5.	活动与执行者间关系 (Performer Relationship) .....	46
7.5.6.	最终期限 (Deadline) .....	47
7.5.7.	过程模拟信息 (Simulation Information) .....	49
7.5.8.	转移约束 (Transition Restrictions) .....	50
7.5.8.1.	Join .....	51
7.5.8.2.	Split .....	52
7.5.9.	一致分类 (Conformance Classes) .....	54
7.6.	转移信息 (Transition Information) .....	54

---

7.6.1.	Condition .....	56
7.6.1.1.	异常转移条件 (Exception Conditions) .....	57
7.7.	工作流参与者 (Workflow Participants) .....	58
7.7.1.	工作流参与者实体类型.....	59
7.8.	工作流相关数据 (Workflow Relevant Data) .....	60
7.9.	数据类型 (Data Type) .....	62
7.9.1.	基本数据类型(Basic Data Type) .....	63
7.9.2.	复杂数据类型 (Complex Data Type) .....	64
7.9.2.1.	Schema Type .....	64
7.9.2.2.	记录类型 (Record Type) .....	65
7.9.2.3.	联合类型 (Union Type) .....	66
7.9.2.4.	枚举类型 (Enumeration Type) .....	66
7.9.2.5.	数组类型 (Array Type) .....	67
7.9.2.6.	列表类型 (List Type) .....	68
7.9.3.	开发者声明的数据类型 (Declared Data Type) .....	68
7.9.3.1.	类型声明 (Type Declaration) .....	68
7.9.3.2.	Declared Type .....	70
8.	工作流例子.....	71
8.1.	定单登记过程.....	71
8.1.1.	EOroder 主过程.....	71
8.1.2.	CreditCheck 子过程.....	72
8.1.3.	FillOrder 子过程.....	72
8.2.	类型声明 (Type Declarations) .....	73
8.3.	ExtendedAttributes .....	76
8.4.	External References .....	77
8.5.	Sample XPDL.....	78
9.	XPDL Schema.....	111

## 2. 对象

这篇文档是面向那些希望实现 workflow 管理联盟(wfmc)的 XML 过程定义语言(XPDL)的组织机构, 也包括那些希望自己产品能够遵守统一规范的开发商。

## 3. 目的

WFMC 为 workflow 系统制定了 5 类功能接口。该文档介绍接口 1——过程定义输入 / 输出接口, 这个接口包含了一个用于描述过程定义的公共元模型, 以及过程定义间进行相互转换的 XML Schema。

## 4. 简介

我们可以用各种不同的工具来分析, 建模, 描述业务过程。而使用 workflow 过程定义接口所定义的公共交换格式, 可以实现两个不同系统间 workflow 过程定义的相互转化。

这个接口为开发和运行环境之间提供了形式上的分离, 可以用建模工具产生过程定义, 并且这个过程定义可以用于多个 workflow 产品。

由建模工具产生的 workflow 过程定义, 可以被不同的 workflow 运行系统所解释。基于公共转换格式的过程定义还可以在这些系统间相互转换, 实现过程定义的分离存储。

为了提供一个公共的方法来访问和描述 workflow 定义, WFMC 制定了一个 workflow 过程定义元数据模型。在元数据模型中给出了过程定义经常用到的一些实体, 并为这些实体制定了各种不同的属性。以元数据模型为基础, 利用公共转换格式, 开发者的特定工具可以对模型进行转换。

使用不同的工具对消息处理进行扩展, 是 XPDL 的一个关键特性。XPDL 不能支持所有工具中的附加消息。依靠一系列有限的描述 workflow 过程定义的实体, XPDL 可以支持一些不同的消息处理方法。

XPDL 是一个通用的框架, 这也是 XPDL 的一个重要特性, 这个框架可以支持开发者的特殊使用目的, 而对外表现形式却相同。开发者的实现和它的外部接口可以独立分开, 因为不管你是如何实现的, 采用什么图形描述(状态图, 活动图等), 只要外部接口符合 XPDL 规范, 那么就可以保持相同的表示形式。希望在 WFMC 接口 1 的未来发布版本中, 能够包含现版本中缺少的属性。

本文档对元模型进行了解释，元模型用来定义过程定义中包含的对象和属性，XPDL 语法与这些对象、属性直接相关，但需要开发者提供以下两种操作：

- 1、 导入一个 XPDL 格式的工作流定义
- 2、 根据开发者对 XPDL 的内部表现形式，输出一个工作流定义

开发者可以使用 XSL 实现这两个操作。

本规范中的关键字与词组均包含在 WfMC 术语表中。在本文中“过程定义”、“业务过程模型”、“工作流模型”都表示相同的概念，可以相互替换。

## 4.1. 一致性

只有通过 WfMC 的权威测试，开发者才可以声明自己的产品是符合 WfMC 规范的，WfMC 通过一定的测试程序来判断一个产品是否符合其所制定的规范。

必须采用 XPDL 语法来实现过程定义导入与输出的一致性，但一定要包含一个最小对象集，这个最小对象集将在本文中给出。

如果建模工具的性能变化幅度很大，可以单独开发一个符合规范的工具，但此工具不能与其他规范的工具进行完全的定义转换。一个符合规范的产品必须能生成有效的、语法正确的 XPDL，并且能解析有效的 XPDL。

## 5. 过程定义内部转换总览

什么是过程定义：表示一种由 workflow 管理系统设定的自动操作为形式的商业过程，过程定义由各种活动以及相互之间的网状关系组成，标识了过程的开始和终止，并且包含了个体行为的信息，比如各个参与者、与 IT 相关的应用程序和数据，等等（WfMC 术语表- WfMC-TC-1011）

过程定义提供了可以对一个过程进行充分描述的环境，可以实现以下功能：

- 1、 在过程制定阶段，可以作为建立和控制过程实例的模板。
- 2、 实现过程的模拟和实现。
- 3、 作为监控和分析制定过程的基础。
- 4、 实现文档编制、可视化和知识管理。

过程定义可能会涉及到子流程，子流程是单独定义的，是整个过程的组成部分。

最初的过程定义必须包含一个最小对象、属性集，这个最小的对象、属性集是过程运行所必须的。这个集合中的一些对象，将被过程的所有实例继承。程实例所继承。

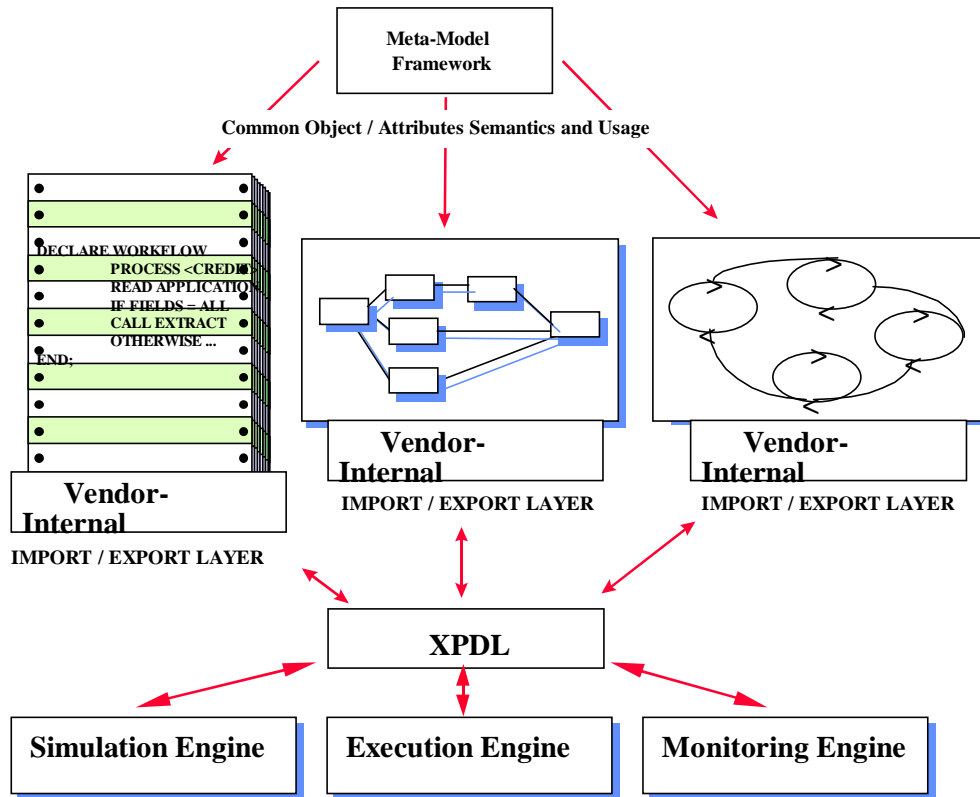
WfMC 术语表也包含了在一个过程定义中基本概念的具体化比如行为、转变、工作流相关数据以及参与者等等。

## 5.1. 过程定义相互转换方法

这个规格使用了 XML 作为过程定义相互转换机制。XPDL 制定了一个公共转换标准，可以实现内部转换格式和公共转换标准相互映射。

根据不同的需求，可以利用不同的机制在系统间进行数据转换。无论什么情况下，过程定义都必须表示成基于公共的对象、关系、属性集的一致性形式。

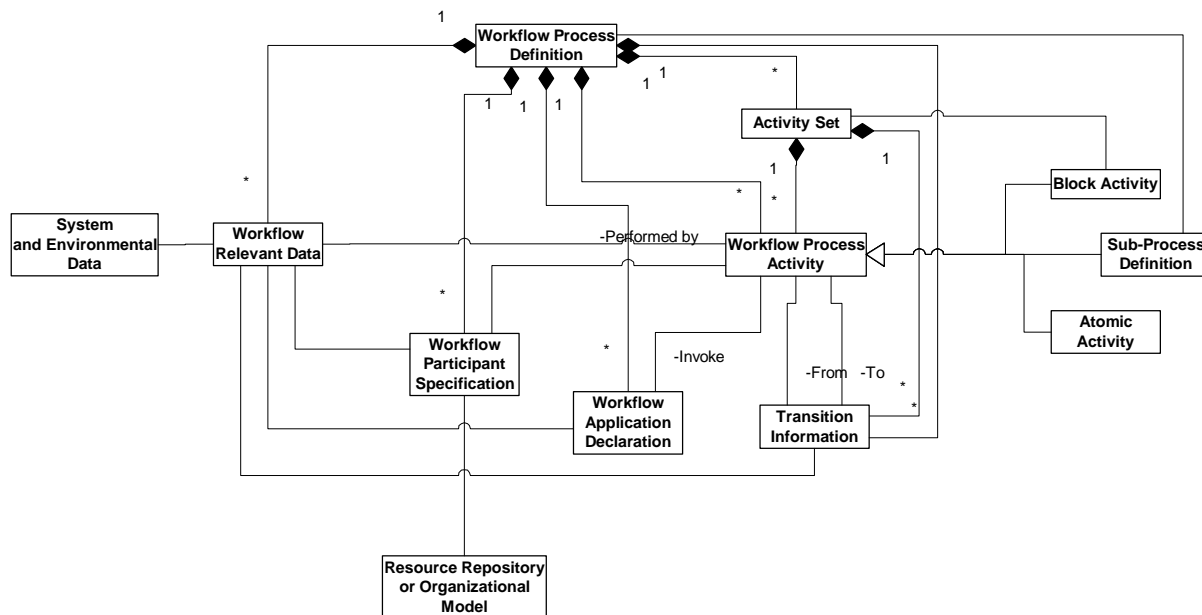
图 5-1 描述了过程定义转换的原理：



## 6. 元模型

元模型描述了过程定义所需要的上层实体，并且为过程定义划分成相应的过程模型、在不同的过程定义与模型中交叉使用公共定义数据提供了多方面的协定。

上层实体由下图所示：



上图中的每个实体，都有相关的属性集与之对应。下面对这些实体及其属性进行详细描述。

## 6.1. 实体总揽

元模型定义了过程定义转换中需要用到的基本实体集。上层实体如下：

### 6.1.1. 工作流过程定义

过程定义实体为过程中其他实体提供上下文信息。过程定义实体是一个过程容器，为过程管理（创建日期、作者等），或者过程执行提供相关数据。

### 6.1.2. 工作流过程活动

过程定义由一个或多个活动组成，并且每个活动都包含整个过程中的一个逻辑的、自包含的工作单元。一个活动代表一个需要由一些相关资源处理，或者由计算机应用程序处理的任任务。有些信息是与活动相关的，例如：活动是否自动由工作流管理系统开始或者结束的信息、当发生资源抢夺时活动的优先权是否高与其他活动的信息。活动使用过程定义相关数据也许需要详细说明。一个活动的范围局限于详细的过程定义。

一个活动也许是一个子流程，在这种情况下，这个活动是一个特定的、分离的过程定义的执行容器，



这个过程可能在相同的工作流服务器上执行，或者在远程服务器上执行（通过使用过程互操作接口）。过程定义允许在子流程中可以包含其自己的活动定义、内部转换、资源和应用程序的分配（即使这些是从父流程中继承而来）。使用输入/输出参数可以实现必要的工作流相关数据在调用与被调用过程间进行交换。（在需要返回参数的时候，返回）

有的活动或许是一个区块活动，负责执行一个活动集，或者对转移与活动进行映射。活动集中的活动、转移共享过程的名称空间。

最后，虚活动是一个活动骨架，不执行任务处理（因此，不需要为其分配资源和应用程序）。但虚活动负责在输入转移、输出转移间进行判断，选择路由。

### 6.1.3. 转移信息

活动间通过流程控制条件（转移信息）相互联系起来。每个转移都有三个基本的属性：源活动、目的活动和转移条件。从一个活动到另一个活动的转移可能需要转移条件（用来判断转移是否可以进行），也可能不需要转移条件。转移可能产生一个连续的活动序列，也可能产生一系列活动的并行执行。一个转移离开一个活动的相关信息（离开动作）以及这个转移进入另一个活动的条件（进入动作），都在相应的活动中进行定义。离开动作是当一个活动要结束、转移要发生时在转移的源活动中被处理，进入动作是当一个转移要结束、目标活动要开始时在转移的目标活动中被处理。通过“离开动作”与“进入动作”，可以把与单线程或同步执行的过程实例的工作流控制，作为相关活动的一部分进行管理，而且还能够保留转移的简单路由分配功能。一个转移的范围局限于包含这个转移及其相关活动的过程定义。

复杂的转移，是不能通过上边提到的三个基本属性以及离开、进入动作表示清楚。复杂的转移需要虚活动来构造，虚活动作为一些活动的中间步骤，而且可以把一些离开动作和进入动作联系在一起。使用附加虚活动的基本转移实体，可以实现任意复杂的路由结构。在工业界中存在几种转移控制方法，因此在 XPD L 中定义了相关的几类转移控制方法。将在下文进行讲述。

### 6.1.4. workflow 参与者 声明

workflow 参与者声明描述了在过程定义中各个活动的执行者所需要的资源。特定的资源可以分配给特殊的活动，并且作为活动的一个属性来进行说明。资源分配把活动与可以为使用的资源集（workflow 参与者声明中描述的资源）联系起来。当资源是一个人时，workflow 参与者声明不是必须的，但如果使用却能描述人的技能、责任等属性，或者表示使用的资源是机器自动控制的，而不是由人来控制。元模型包含了一些简单的资源类型，这些资源也许能在 workflow 参与者声明中用到。

### 6.1.5. 资源库

资源库中的记录可以是人员、程序或者机器等。在较多情况下，参与者声明就是指资源库。在参与者是人的情况下，资源库也许就是一个组织结构图。注意：本规范中没有定义一个资源库。

### 6.1.6. workflow应用程序声明

workflow应用程序声明描述了workflow服务需要调用的IT应用程序或者接口。workflow服务调用这些应用程序和或者接口来支持或者完全自动的对相关活动进行处理，这些应用程序以及接口在活动的应用程序属性中定义。被调用的应用程序可以是工业工具软件、特殊部门（企业）服务、或者在workflow管理系统中实现的本地过程（方法）。workflow外部应用程序声明反映了workflow机与外部应用程序之间的接口，包括任何传递参数。

### 6.1.7. workflow相关数据

workflow相关数据是指那些在过程运行期间，每个过程实例都需要用到或者创建的数据。活动或者过程在执行时必须得到这些数据，这些数据也可以用来在活动间传递持久信息或者中间结果。在条件表达式中也可能需要这些数据，例如，在转移和参与者分配中。workflow相关数据是一种特殊的数据类型。XPDL中定义一些基本的和负责的数据类型（例如，date, String等）。活动、被调用的程序以及转移条件都可能涉及到workflow相关数据。

### 6.1.8. 系统和运行环境数据

系统和运行环境数据是由workflow管理系统或者本地系统保存的，但是这些数据可以被workflow活动反问，被workflow管理系统中的条件表达式所使用，就如同workflow相关数据一样。

### 6.1.9. 数据类型和表达式

元模型中给出了一些标准的数据类型（string, reference, integer, float, date/time, 等），这些数据类型与workflow相关数据、系统/运行数据或者参与者数据相关。需要使用这些标准的数据类型来构造表达式，从而进行条件计算。可以使用XML Schema 或者使用外部资源中的参考数据定义，来对标准数据类型进行扩充。

## 6.2. 过程、包

在上图中，过程模型中所包含的实体，其使用范围也许比单一的过程定义要广。尤其是 workflow 参与者定义、应用程序定义、和 workflow 相关数据定义也许要被多个 workflow 定义所使用。元模型采用通用的过程定义资源库来保存各种构成过程定义的实体类型。在资源库内部，为了有效的把过程定义数据导入、导出资源库，提出了包的概念。包就象一个容器，对许多不同的过程定义中的通用数据实体进行分组，从而避免在每一个单独的过程定义中重复定义那些数据。

包中存放了许多 workflow 过程定义实体的公共属性（作者、版本、状态等）。包中的每一个过程定义都自动继承包中的公共属性，除非在过程定义中那些属性被个别的重新定义。

在包中，有些实体的定义范围是全程的，并且这些实体能被同一包中的任何 workflow 定义所引用。这些实体为：

- 1、 workflow 参与者描述
- 2、 workflow 外部应用程序声明
- 3、 workflow 相关数据

包引用允许使用包内对象或者所引用的外部包中的上层实体对象：

- 1、 子流程可以引用过程 ID 号
- 2、 workflow 参与者声明
- 3、 workflow 外部应用程序声明

在同一 workflow 定义库中不同的包间使用名称、标识符管理约定，开发者可以得到必须唯一的名称以及标识符。在过程制定阶段，名称引用按照如下默认的约定顺序进行搜索：

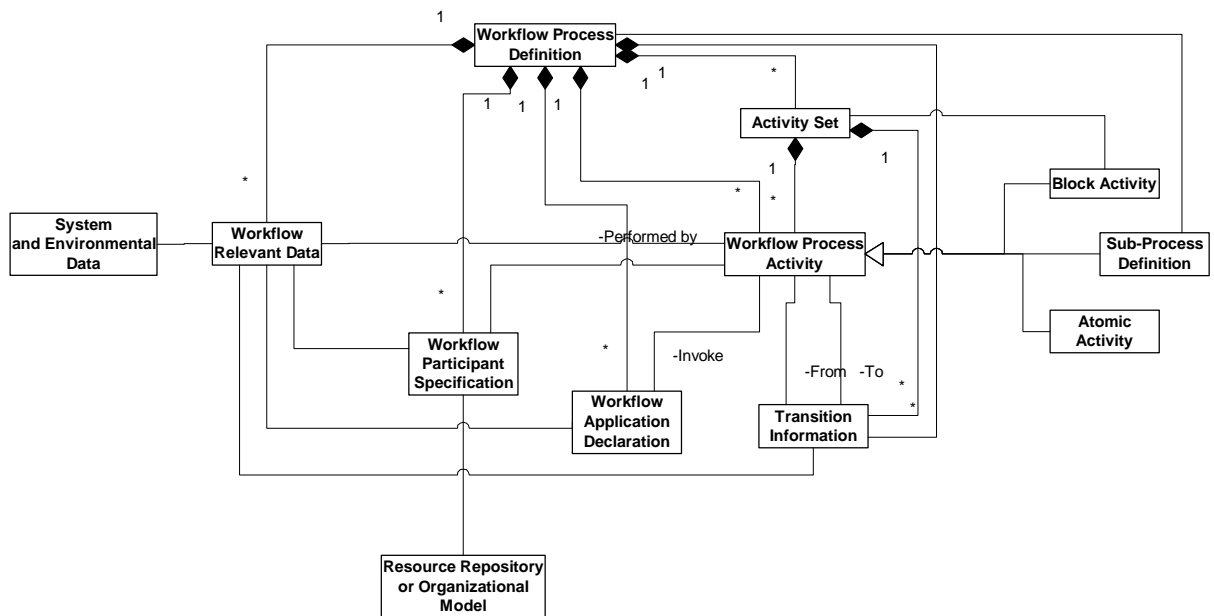
- 过程 ID 号 —— 先在同一个模型内进行搜索（包括在不同服务机上远程执行的参考过程定义），然后是外部参考模型。
- 外部应用程序/workflow 参与者 —— 先在同一个模型中搜索，然后在外部参考模型中搜索。

workflow 相关数据的命名在一个包中必须唯一。workflow 相关数据作为参数在过程间进行传递的地方，本版规范中将使用语义学的约定。过程设计者 / 管理者的责任是确保过程定义中、模型中的名称、数据类型保持一致，以便支持子流程运行。

### 6.3. 过程元模型

元模型为 workflow 定义转换定义了基本的实体集和属性集。每一个过程定义中，下边列出的实体都必须被定义，或者在过程定义时明确给出、或者直接从包中继承、或者通过对包的交叉引用。

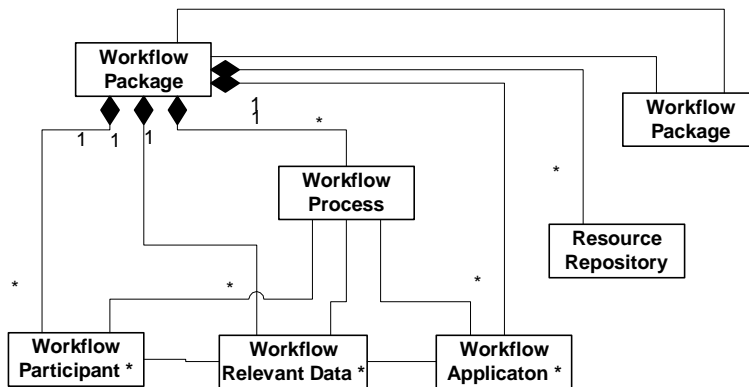
- 工作流过程活动
- 转移信息
- 工作流参与者声明
- 工作流外部应用程序声明
- 工作流相关数据



## 6.4. 包元模型

在一个模型定义中，多个过程定义被绑定到一起。包作为一个容器，来对单独的过程定义及其相关实体数据进行分组，这些数据适合这个过程中所包含的所有过程（因此，这些数据只需要定义一次）。包元模型中包含以下实体类型：

- 工作流过程定义
- 工作流参与者声明
- 工作流外部应用程序声明
- 工作流相关数据



\* entities can be redefined in the Workflow Process

图 6 - 5 包定义元模型

包元模型为过程模型的转换、存储定义了实体、属性。元模型为参与者声明、外部应用程序声明、工作流相关数据定义了一些继承规则，来把单独的过程定义与实体定义联系起来，可以在包层次级中对那些实体进行定义，而不是在单独的过程定义级上。

包定义允许对一系列的公共过程定义属性进行声明，这些属性声明可以用在这个包下的所有单独的过程定义中。从而在单独的过程定义中可以不对这些属性进行定义。（如果，在单独的过程定义层中对这些属性进行了重定义，那么在这个过程中将采用重定义了属性，而不是包层上进行定义的属性）

### 6.4.1. 过程库

过程定义导入 / 导出接口，在与 workflow 管理系统联系的工作流定义库间进行工作。通过把符合 XPDL 格式的文件与过程定义库间进行传递来实现此接口。本接口规范允许在单独过程定义级和包层级上进行过程定义数据导入与导出。

过程库与 workflow 控制模块间的内部接口，不是本规范中的一部分，是由开发者的来自定义的。内部接口提供对过程库的分离使用：静态过程库（持久、运行存储过程定义数据）、动态使用（管理过程实例运行时的变化）。

过程定义库的本地存储结构不是 WfMC 规范的一部分。使用包只是为了方便可复用数据的导入、导出。当使用一个简单的、为单个过程定义服务的过程定义库时，输入包中的公共信息也许会被每个单独的过程定义通过导入接口来复制。

#### 6.4.1.1. 重定义和范围

可能需要重新定义属性、元模型实体、参考外部包，所以在 XPDL 结构中引入了范围和层次的概念：

##### ① 工作流相关数据

工作流过程相关数据是有范围的，其范围是由直接所属的元模型实体定义的，并且不能是嵌套的。标识符的可见性也是被那个实体定义的。

##### ② 属性

属性（包括外部属性）也是有范围的，其范围是由直接所属的元模型实体定义的，并且是嵌套的，例如，可以在低层进行重定义：名称属性要在每一个实体定义中重新定义。扩展属性标识符的在个别实体内以及所有子实体是可见的，除非在子实体中标识符被重新定义了。

##### ③ 工作流参与者和外部应用程序

工作流参与者、外部应用程序与扩展属性有相同的范围和可见性。所有引用的工作流相关数据和扩展属性，在其使用范围内（至少在同一个包中），都必须定义。

对一个引用的外部包实体，需要参考在外部包中定义的实体和标识符，条款机制以外部包中的根路径开始。这样就能够保证在调用过程，调用有相同 ID 的实体时，不会发生冲突。

## 6.5. 元素总揽

下表中给出来 XPDL 中定义的主要元素的简要介绍：

- 第一行包括所有主要元素的通用属性和元素。主要元素都有属性 ID、名称、有可能有描述和扩展属性。
- 第二行是每个主要元素的特有属性。
- 第三行的元素是需要参考其他元素的。
- 文档和图表元素包含执行机所需要的描述信息。
- 最后一行是过程模拟、过程优化的相关信息。

更多的元素和预定义的属性可能被加到模型中，来创建将来一致的模型。所有元素的描述和语义将在后边的章节中给出。

Package	Workflow Process	Activity	Transition	Application	Data Field (Workflow Relevant Data)	Participant
- Id	- Id	- Id	- Id	- Id	- Id	- Id
- Name	- Name	- Name	- Name	- Name	- Name	- Name
- Description	- Description	- Description	- Description	- Description	- Description	- Description
- Extended Attributes	- Extended Attributes	- Extended Attributes	- Extended Attributes	- Extended Attributes	- Extended Attributes	- Extended Attributes
- XPDL Version	- Creation Date	- Automation Mode			- Data Type	- Participant Type
- Source Vendor ID	- Version	- Split				
- Creation Date	- Author	- Join				
- Version	- Codepage	- Priority				
- Author	- Country Key	- Limit				
- Codepage	- Publication Status	- Start Mode				
- Country Key	- Priority	- Finish Mode				
- Publication Status	- Limit	- Deadline				
- Conformance Class	- Valid From Date					
- Priority Unit	- Valid To Date					
- Responsible	- Parameters	- Performer	- Condition	- Parameters	- Initial Value	
	- Responsible	- Tool	- From			
		- Subflow	- To			
		- ActivitySet				
		- Actual Parameter				
- External Package						
- Documentation	- Documentation	-Documentation-				
- Icon	- Icon	-Icon				
- Cost Unit	-Duration Unit	-Cost				

	- Duration	- Duration				
	- Waiting Time	- Waiting Time				
	- Working Time	- Working Time				

### 6.5.1. 开发者扩展元素

尽管元模型和 XPD 包含了很多在过程定义转换中需要的结构，但是在某些特殊情况下，过程定义中需要一些另外的信息。开发者被鼓励尽可能用标准的实体/属性集进行过程定义。下边描述的支持扩展机制，为表达交换意图提供了一种标准的方法，但在过程制定阶段，需要本地系统为运行提供支持。

#### 6.5.1.1. 扩展属性

扩展属性是由开发者定义的属性，用来描述需要在系统间交换的实体特性。在过程制定阶段使用的扩展属性，在运行阶段的语义是单独定义的，并且过程定义的导出和工作流服务器导入方需要达成一致的协议。

#### 6.5.1.2. 扩展参数映射

在这篇规范中，没有对编译和参数传递的详细 Scheme 进行定义。使用工作流互操作规范（接口 4），来实现远程子流程调用中的参数传递，在互操作的具体语法规范中，给出了这些参数映射的规则。参数映射和编译的局部 Scheme，都是由开发者自己定义的。



## 7. XML 过程定义语言

### 7.1. 多实体通用元素

#### 7.1.1. 扩展属性

所有实体都可以使用扩展属性。开发者通过扩展属性来对本篇规范的功能进行扩充，从而满足自己产品的需要。开发者可以增加子内容到 `ExtendedAttribute` 中：

```
<xsd:element name="ExtendedAttribute">
  <xsd:complexType mixed="true">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
      <xsd:any minOccurs="0" maxOccurs="unbounded"/>
    </xsd:choice>
    <xsd:attribute name="Name" type="xsd:NMTOKEN" use="required"/>
    <xsd:attribute name="Value" type="xsd:string"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="ExtendedAttributes">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="xpd1:ExtendedAttribute"
        minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

	描 述
--	-----

	描 述
Name	用来标识扩展属性
Value	属性值

### 7.1.2. 形式参数

在 workflow 定义和 workflow 应用程序中，形式参数可以作为属性来使用。形式参数在调用时被传递，并且返回控制句柄。

```
<xsd:element name="FormalParameter">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="xpdl:DataType" />
      <xsd:element ref="xpdl:Description" minOccurs="0" />
    </xsd:sequence>
    <xsd:attribute name="Id" type="xsd:NMTOKEN" use="required" />
    <xsd:attribute name="Index" type="xsd:NMTOKEN" />
    <xsd:attribute name="Mode" default="IN">
      <xsd:simpleType>
        <xsd:restriction base="xsd:NMTOKEN">
          <xsd:enumeration value="IN" />
          <xsd:enumeration value="OUT" />
          <xsd:enumeration value="INOUT" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:attribute>
  </xsd:complexType>
</xsd:element>

<xsd:element name="FormalParameters">
```

```

<xsd:complexType>
  <xsd:sequence>
    <xsd:element ref="xpd1:FormalParameter"
      minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
</xsd:element>

```

	描 述
Data type	形式参数的数据类型。
Description	形式参数的文本描述
Id	形式参数的标识
Index	参数的索引
Mode	IN            输入参数 OUT          输出参数 INOUT        输入、输出参数

#### 7.1.2.1. 参数传递语义

参数传递语义定义如下：

- i. 在调用中，只读的形式参数（IN）由其相对应的实际参数的值初始化。这就是按值传递语义。
- ii. 可读/写形式参数（INOUT）由其相对应的实际参数的值来初始化，但实际参数必须是 workflow 相关数据实体的标识符。当过程完成后，输出形式参数的值要复给对其初始化的实际参数。这是拷贝归还语义。
- iii. 只写形式参数（OUT）初始化为 0（String 类型的被设为空）。在过程完成后，输出形式参数的值要复给原始的实际参数（实际参数必须是 workflow 相关数据实体的标识符）。这就是 0-归还语义。

#### 7.1.2.2. 并发语义

参数的拷贝和存储是自动完成的，从而避免在并发的 workflow 相关数据操作中出现访问冲突，而这些操作是序列化的。在 7.1.2.1 的 (C) 中，调用返回时，拷贝和存储间默认是不加锁的，并且返回参数值会覆盖本地值。

### 7.1.2.3. 形式——实际参数 映射

参数映射表定义了，调用过程中实际参数向形式参数的映射。实际参数按照顺序 1: 1 映射到形式参数，例如第一个实际参数映射到第一个形式参数，第二个实际参数映射到第二个形式参数。在过程定义时需要类型兼容，或者在 workflow 系统运行时进行强制类型转换。

在实际参数是一个表达式的情况下，工作流机对表达式进行计算和缓冲，同时缓冲器的内容用来进行形式——实际参数映射。缓冲和运算是如何运行的超出了本篇规范的范围。

### 7.1.3. 外部参考

外部参考是一个实体的外部定义参考。它可以用于数据类型、参与者以及应用程序声明中。

```
<xsd:element name="ExternalReference">
  <xsd:complexType>
    <xsd:attribute name="xref" type="xsd:NMTOKEN" use="optional"/>
    <xsd:attribute name="location" type="xsd:anyURI" use="required"/>
    <xsd:attribute name="namespace" type="xsd:anyURI" use="optional"/>
  </xsd:complexType>
</xsd:element>
```

	描 述
Location	指明类型的文档的 URI
Namespace	实体定义的有效范围
Xref	对在外部文档中实体进行标识

例 1、用 XML schema 定义的一个形式参数：

```
<FormalParameter Id="PO">
```

```
<DataType>
    <ExternalReference location="http://abc.com/schemas/po.xsd"/>
</DataType>
<Description>PO specification for abc.com</Description>
</FormalParameter>
```

例 2、用 Java class 定义一个 DataField

```
<DataField Id="PO" Name="PurchaseOrder" IsArray="FALSE">
    <DataType>
        <ExternalReference location="com.abc.purchases.PO"/>
    </DataType>
    <Description>PO specification for abc.com</Description>
</DataField>
```

### 7.1.3.1. *Web Services*

过程的活动也许要调用 web 服务，ExternalReference 元素可以作为在 Web Service (WSDL) 文档中定义的应用程序、数据类型的参考。

例 3、一个 DataField 其数据类型在 WSDL 文档中定义：

```
<DataField Id="abcPO" Name="abcPurchaseOrder" IsArray="False">
    <DataType>
        <ExternalReference xref="PO"
            location="http://abc.com/services/poService.wsdl"
            namespace="poService/definitions/types"/>
    </DataType>
</DataField>
```

例 4、在 WSDL 文档中定义为一个操作的应用程序

```
<Application Id="placeOrder">
  <ExternalReference
    location="http://abc.com/PO/services/poService.wsdl"
    xref="PlaceOrder"
    namespace=
      "http://abc.com/services/poService.wsdl/definitions/portType" />
</Application>
```

## 7.2. 包定义

可以在一个包中定义多个过程，这些过程可以共享相同的工具和参与者。我们建议，为每个业务过程创建一个包，而且包中应该包含所有的必要工作流过程、相关工具和工作流参与者。也可以在一个包中只定义一个过程定义的一部分，或者定义几个过程的公共部分（例如，工作流参与者列表和应用程序列表）：

```
<xsd:element name="Package">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="xpdl:PackageHeader" />
      <xsd:element ref="xpdl:RedefinableHeader" minOccurs="0" />
      <xsd:element ref="xpdl:ConformanceClass" minOccurs="0" />
      <xsd:element ref="xpdl:Script" minOccurs="0" />
      <xsd:element ref="xpdl:ExternalPackages" minOccurs="0" />
      <xsd:element ref="xpdl:TypeDeclarations" minOccurs="0" />
      <xsd:element ref="xpdl:Participants" minOccurs="0" />
      <xsd:element ref="xpdl:Applications" minOccurs="0" />
      <xsd:element ref="xpdl>DataFields" minOccurs="0" />
      <xsd:element ref="xpdl:WorkflowProcesses" minOccurs="0" />
      <xsd:element ref="xpdl:ExtendedAttributes" minOccurs="0" />
    </xsd:sequence>
    <xsd:attribute name="Id" type="xsd:NMTOKEN" use="required" />
  </xsd:complexType>
</xsd:element>
```

```

        <xsd:attribute name="Name" type="xsd:string"/>
    </xsd:complexType>
</xsd:element>

```

	描 述
Applications	工作流应用程序声明列表
Conformance Class	在此包中过程定义的结构限制
Data Fields	此包的工作流相关数据列表
Extended Attributes	或许要加入到包中的开发者自定义的扩展属性
External Packages	参考在其他文档中定义的包定义
Id	包 ID 号
Name	用来标识包的文本
Package Header	描述包特性的元素集合
Participants	用来实现包中过程的资源列表
Redefinable Header	包和过程定义中都使用的元素、属性集合
Script	表达式中所使用的脚本语言
Type Declarations	包中用到的数据类型列表
Workflow Processes	组成包的工作流过程列表

### 7.2.1. 包定义头 (Package define header)

包定义头包含包的所有重要信息，例如，XPDL 的版本，源开发者 ID 等。

```

<xsd:element name="PackageHeader">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="xpd1:XPDLVersion"/>
            <xsd:element ref="xpd1:Vendor"/>
            <xsd:element ref="xpd1:Created"/>
            <xsd:element ref="xpd1:Description" minOccurs="0"/>
            <xsd:element ref="xpd1:Documentation" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>

```

```

        <xsd:element ref="xpd1:PriorityUnit" minOccurs="0"/>
        <xsd:element ref="xpd1:CostUnit" minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>
</xsd:element>

<xsd:element name="XPDLVersion" type="xsd:string"/>

<xsd:element name="Vendor" type="xsd:string"/>

<xsd:element name="Created" type="xsd:string"/>

<xsd:element name="Description" type="xsd:string"/>

<xsd:element name="Documentation" type="xsd:string"/>

<xsd:element name="PriorityUnit" type="xsd:string"/>

<xsd:element name="CostUnit" type="xsd:string"/>

```

	描 述
Cost Unit	模拟数据所使用单元(通常用流量来描述)
Created	包定义的创建日期
Description	包的文本描述
Documentation	操作系统的路径, 帮助文件以及描述文件的文件名
Priority Unit	表达用户自定义语义的字符串
Vendor	说明模型定义的由来, 包括开发者的名字, 产品名称和产品的发布序号
XPDL Version	XPDL 规范版本, 本规范的现行版本 "0.02".

### 7.2.2. Redefinable Header



可重定义头（Redefinable Header）中的属性，是那些在工作流定义头中或者过程定义头中，可能被重新定义的属性。在从新定义的情况下，要遵守范围规则。

```
<xsd:element name="RedefinableHeader">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="xpdl:Author" minOccurs="0"/>
      <xsd:element ref="xpdl:Version" minOccurs="0"/>
      <xsd:element ref="xpdl:Codepage" minOccurs="0"/>
      <xsd:element ref="xpdl:Countrykey" minOccurs="0"/>
      <xsd:element ref="xpdl:Responsibles" minOccurs="0"/>
    </xsd:sequence>
    <xsd:attribute name="PublicationStatus">
      <xsd:simpleType>
        <xsd:restriction base="xsd:NMTOKEN">
          <xsd:enumeration value="UNDER_REVISION"/>
          <xsd:enumeration value="RELEASED"/>
          <xsd:enumeration value="UNDER_TEST"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:attribute>
  </xsd:complexType>
</xsd:element>

<xsd:element name="Author" type="xsd:string"/>
<xsd:element name="Version" type="xsd:string"/>
<xsd:element name="Codepage" type="xsd:string"/>
<xsd:element name="Countrykey" type="xsd:string"/>
<xsd:element name="Responsible" type="xsd:string"/>
<xsd:element name="Responsibles">
  <xsd:complexType>
    <xsd:sequence>
```

```

        <xsd:element ref="xpdl:Responsible" minOccurs="0"
maxOccurs="unbounded" />
    </xsd:sequence>
</xsd:complexType>
</xsd:element>

```

	描 述
Author	包定义的作者
Code page	正文部分的页代码
Country key	以 ISO 3166 基础的国家代码，或者是 3 位的数字国家代码，或者是 2 位的字母代码
Publication Status	工作流过程定义（WorkFlow Process Define）的状态  UNDER_REVISION  RELEASED  UNDER_TEST
Responsible(s)	对此工作流过程负责的工作流参与者；工作流运行时期的监督人。  与工作流参与者相联系。  工作流参与者（通常是有组织的机构或者人）对本模型定义中的工作流负责。  一般认为运行时期的监督人是值得信赖的。默认是：初使参与者。
Version	包定义的版本

### 7.2.3. 一致类声明（Conformance Class Declaration）

一致类声明使得在模型定义中的一致类描述受到限制。被说明的类适用于包中所有的过程定义，除非在过程定义层上进行了重新定义。

```

<xsd:element name="ConformanceClass">
    <xsd:complexType>
        <xsd:attribute name="GraphConformance">
            <xsd:simpleType>
                <xsd:restriction base="xsd:NMTOKEN">
                    <xsd:enumeration value="FULL_BLOCKED" />
                    <xsd:enumeration value="LOOP_BLOCKED" />
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:attribute>
    </xsd:complexType>
</xsd:element>

```

```

        <xsd:enumeration value="NON_BLOCKED" />
    </xsd:restriction>
</xsd:simpleType>
</xsd:attribute>
</xsd:complexType>
</xsd:element>

```

	描 述	
Conformance Class	FULL-BLOCKED	构成的网络结构不能是 SPLIT/JOIN 嵌套或者循环
	LOOP-BLOCKED	类构成的网络结构不能是循环结构
	NON-BLOCKED	对结构没有限制, 此为默认值

#### 7.2.4. Script

用 Script 元素指定在 XPD L 表达式中可以适用的脚本语言。文本表达式可能会用在每一个类型是 xsd:String 的元素中。例如, 可以在 ActualParameter 或者 Cost 元素中适用一个文本表达式。

符合 XML 格式的表达式可以用在 Xpression 元素中 (例如用在转移条件中)。

```

<xsd:element name="Script">
    <xsd:complexType>
        <xsd:attribute name="Type" type="xsd:string" use="required"/>
        <xsd:attribute name="Version" type="xsd:string" use="optional"/>
        <xsd:attribute name="Grammar" type="xsd:anyURI" use="optional"/>
    </xsd:complexType>
</xsd:element>

```

	描 述
Type	明确在表达式中适用的脚本语言。为了实现的一致, 推荐使用以下字符串来定义使用的脚本语言: text/javascript, text/vbscript, text/tcl, text/ecmascript, text/xml
Version	脚本语言的版本
Grammar	明确语言文法的参考文本。例如: XML schema, DTD, BNF

### 7.2.5. 外部包参考 (External Package Reference)

外部包参考为工作流系统提供了一个接口，可以参考其他包或者其他系统中的定义。

```
<xsd:element name="ExternalPackage">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="xpdl:ExtendedAttributes" minOccurs="0"/>
    </xsd:sequence>
    <xsd:attribute name="href" type="xsd:string"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="ExternalPackages">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="xpdl:ExternalPackage"
        minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

	描 述
Extended Attributes	可选的、开发者自定义的属性，用来满足特殊需求
href	模型标识，模型的逻辑参考

### 7.3. 工作流应用程序声明 (Workflow Application Declaration)

工作流应用程序声明，是工作流过程定义中或者其所在包中的工作流过程需要或调用的应用程序、工具的列表。工具的定义也许就是一个名字而已。这意味着，真正的工具定义不是必要的，并且可能是由对象管理器来处理。这样的好处是，可以处理多平台环境。XPDL 抽象于具体的运行环境和实现环境。

```

<xsd:element name="Application">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="xpd:Description" minOccurs="0"/>
      <xsd:choice>
        <xsd:element ref="xpd:FormalParameters"/>
        <xsd:element ref="xpd:ExternalReference"
minOccurs="0"/>
      </xsd:choice>
      <xsd:element ref="xpd:ExtendedAttributes" minOccurs="0"/>
    </xsd:sequence>
    <xsd:attribute name="Id" type="xsd:NMTOKEN" use="required"/>
    <xsd:attribute name="Name" type="xsd:string"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="Applications">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="xpd:Application" minOccurs="0"
maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

```

	描 述
Description	应用程序简短描述
Extended Attributes	可选的、开发者自定义的属性，用来满足特殊需求
External Reference	外部应用程序签名规范的参考
Formal Parameters	通过调用接口与应用程序交互的参数列表。
Id	工作流应用程序定义 ID

	描 述
Name	指明一个应用程序文本(可以被解释为工具的总名称)

### 7.3.1. 调用参数 (Invocation Parameters)

工作流应用程序声明中可能要对参数进行定义，并且在其他实体中使用这些参数定义。

## 7.4. 工作流过程定义 (Workflow Process Define)

工作流过程定义，详细说明了构成工作流的元素。包含活动、转移、应用程序和过程相关数据实体的定义或者声明。属性可以作为管理相关数据来被说明，例如作者、版本；也可能作为运行时数据来说明，例如优先级；也可作为 BPR 或者工作流过程模拟相关数据来说明。

一个工作流过程可以作为一个典型子流的实现来运行。在这种情况下，参数可能会定义为过程的属性。

如果工作流过程定义中包含输入参数，并且参数不是通过子流程调用方法来初始化的，那么这样的工作流定义是本地化的。在这样的环境中，与过程定义实例有关的、在参数表中的所有工作流相关数据都要被初始化成默认值。如果工作流相关数据不作为输入参数来传递或者不被初始化成默认值，那么结果是不确定的。同样，当一个子流程非法终止，没有返回任何参数值给调用过程，结果也是不能确定的。

通常，实体的标识符和名称在其所在的实体范围内有效。标识符在其有效范围内必须是唯一的。对过程的标识符、名称来说其范围是其所在的包。

```
<xsd:element name="WorkflowProcess">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="xpdl:ProcessHeader"/>
      <xsd:element ref="xpdl:RedefinableHeader" minOccurs="0"/>
      <xsd:element ref="xpdl:FormalParameters" minOccurs="0"/>
      <xsd:group ref="xpdl:DataTypes"/>
      <xsd:element ref="xpdl>DataFields" minOccurs="0"/>
      <xsd:element ref="xpdl:Participants" minOccurs="0"/>
      <xsd:element ref="xpdl:Applications" minOccurs="0"/>
      <xsd:element ref="xpdl:ActivitySets" minOccurs="0"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

```

        <xsd:element ref="xpd1:Activities" minOccurs="0"/>
        <xsd:element ref="xpd1:Transitions" minOccurs="0"/>
        <xsd:element ref="xpd1:ExtendedAttributes" minOccurs="0"/>
    </xsd:sequence>
    <xsd:attribute name="Id" type="xsd:NMTOKEN" use="required"/>
    <xsd:attribute name="Name" type="xsd:string"/>
    <xsd:attribute name="AccessLevel">
        <xsd:simpleType>
            <xsd:restriction base="xsd:NMTOKEN">
                <xsd:enumeration value="PUBLIC"/>
                <xsd:enumeration value="PRIVATE"/>
            </xsd:restriction>
        </xsd:simpleType>
    </xsd:attribute>
</xsd:complexType>
</xsd:element>

<xsd:element name="WorkflowProcesses">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="xpd1:WorkflowProcess"
                minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>

```

	描 述
AccessLevel	过程的访问级别或者是 PUBLIC、或者是 PRIVATE。如果是 PUBLIC：过程能被外部系统或者应用程序调用；PRIVATE：过程只能被子流程调用
Activities	组成过程的活动列表。
ActivitySets	活动、转移的自包含集合列表
Applications	工作流应用程序声明列表

	描 述
Data Fields	工作流相关数据列表.
Extended Attributes	可选的、开发者自定义的属性, 用来满足特殊需求
Formal Parameters	传递给过程的参数的列表.
Id	工作流过程 ID
Name	工作流过程名
Participants	用来实现过程的, 资源列表
Process Header	说明过程特性的元素集
Redefinable Header	包和过程定义都可以使用的元素、属性集
Transitions	转移列表

#### 7.4.1. 工作流过程定义头 (Workflow Process Define Header)

工作流过程头保存过程定义的详细信息, 例如过程版本、优先级、有效时间等。

```
<xsd:element name="ProcessHeader">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="xpd:Created" minOccurs="0"/>
      <xsd:element ref="xpd:Description" minOccurs="0"/>
      <xsd:element ref="xpd:Priority" minOccurs="0"/>
      <xsd:element ref="xpd:Limit" minOccurs="0"/>
      <xsd:element ref="xpd:ValidFrom" minOccurs="0"/>
      <xsd:element ref="xpd:ValidTo" minOccurs="0"/>
      <xsd:element ref="xpd:TimeEstimation" minOccurs="0"/>
    </xsd:sequence>
    <xsd:attribute name="DurationUnit">
      <xsd:simpleType>
        <xsd:restriction base="xsd:NMTOKEN">
          <xsd:enumeration value="Y"/>
          <xsd:enumeration value="M"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:attribute>
  </xsd:complexType>
</xsd:element>
```



```

        <xsd:enumeration value="D"/>
        <xsd:enumeration value="h"/>
        <xsd:enumeration value="m"/>
        <xsd:enumeration value="s"/>
    </xsd:restriction>
</xsd:simpleType>
</xsd:attribute>
</xsd:complexType>
</xsd:element>

<xsd:element name="Created" type="xsd:string"/>
<xsd:element name="Description" type="xsd:string"/>
<xsd:element name="Limit" type="xsd:string"/>
<xsd:element name="Priority" type="xsd:string"/>
<xsd:element name="TimeEstimation">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="xpdl:WaitingTime" minOccurs="0"/>
            <xsd:element ref="xpdl:WorkingTime" minOccurs="0"/>
            <xsd:element ref="xpdl:Duration" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>

<xsd:element name="WaitingTime" type="xsd:string"/>
<xsd:element name="WorkingTime" type="xsd:string"/>
<xsd:element name="Duration" type="xsd:string"/>
<xsd:element name="ValidFrom" type="xsd:string"/>
<xsd:element name="ValidTo" type="xsd:string"/>

```

	描 述
Created	工作流过程定义创建日期

	描 述
Description	过程简短描述
Duration	执行一个任务的所用预计时间单元。
Duration Unit	默认时间单元，用整数来计算，可能的时间单元：  Y - 年  M - 月  D - 日  H - 小时  m - 分钟  s - 秒
Limit	期望的过程持续时间，从过程开始计算。达到限制时间（Limit time）的后果在本篇文章中没有说明。在此情况下，至少正运行过程要知道所用时间的情况
Priority	过程优先级  缺省值：从模型定义中继承。
Time Estimation	等待时间（waiting time）、工作时间（working time）、持续时间（duration）。用于过程模拟。
Valid From	工作流过程定义的激活日期，空值代表系统日期。  缺省值：从模型定义中继承。
Valid To	过程定义的开始有效的日期，空值代表一直有效，缺省值：从模型定义中继承。
Waiting Time	准备执行一个任务所用的时间（time estimation），等待时间（Waiting Time）被分析系统环境所占用，并且可能会被运行环境更新
Working Time	执行一个活动完成相关任务所用的时间(time estimation)

#### 7.4.2. 工作流过程可重定义头（Workflow Process Redefinable Header）

```
<xsd:element name="RedefinableHeader">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="xpdl:Author" minOccurs="0"/>
      <xsd:element ref="xpdl:Version" minOccurs="0"/>
      <xsd:element ref="xpdl:Codepage" minOccurs="0"/>
      <xsd:element ref="xpdl:Countrykey" minOccurs="0"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

```

        <xsd:element ref="xpd1:Responsibles" minOccurs="0"/>
    </xsd:sequence>
    <xsd:attribute name="PublicationStatus">
        <xsd:simpleType>
            <xsd:restriction base="xsd:NMTOKEN">
                <xsd:enumeration value="UNDER_REVISION"/>
                <xsd:enumeration value="RELEASED"/>
                <xsd:enumeration value="UNDER_TEST"/>
            </xsd:restriction>
        </xsd:simpleType>
    </xsd:attribute>
</xsd:complexType>
</xsd:element>

<xsd:element name="Author" type="xsd:string"/>
<xsd:element name="Codepage" type="xsd:string"/>
<xsd:element name="Countrykey" type="xsd:string"/>
<xsd:element name="Responsible" type="xsd:string"/>
<xsd:element name="Responsibles">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="xpd1:Responsible" minOccurs="0"
maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
<xsd:element name="Version" type="xsd:string"/>

```

	描 述
Author	工作流过程定义的作者
Codepage	正文部分的页代码 默认值：从模型定义中继承

	描 述
Country key	以 ISO 3166 基础的国家代码，或者是 3 位的数字国家代码，或者是 2 位的字母代码 默认值：从模型定义中继承
Publication Status	工作流过程定义的状态 默认值：从模型定义中继承  UNDER_REVISION  RELEASED  UNDER_TEST
Responsible(s)	对此工作流过程负责的工作流参与者；工作流运行时期的监督人。 与工作流参与者相联系。  工作流参与者（通常是有组织的机构或者人）对本模型定义中的工作流负责。 一般认为运行时期的监督人是值得信赖的。 默认值：从模型定义中继承
Version	工作流过程定义的版本

### 7.4.3. 活动集 (Activity Set)

活动集是活动、转移的自包含集合。一个集合中的转移只能与本集合中的活动相关联。转移不能从一个集合发到另一个集合，也不能有转移指向一个集合。活动集合可以使用块活动来执行。

```

<xsd:element name="ActivitySet">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="xpdl:Activities" minOccurs="0"/>
      <xsd:element ref="xpdl:Transitions" minOccurs="0"/>
    </xsd:sequence>
    <xsd:attribute name="Id" type="xsd:NMTOKEN" use="required"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="ActivitySets">
  <xsd:complexType>

```

```

<xsd:sequence>
    <xsd:element ref="xpdl:ActivitySet" minOccurs="0"
maxOccurs="unbounded" />
</xsd:sequence>
</xsd:complexType>
</xsd:element>

```

	描述
Activities	组成过程的活动列表
Id	工作流过程 ID
Transitions	与过程活动相连的转移列表

## 7.5. 工作流过程活动 (Workflow Process Activity)

工作流活动定义用来说明构成工作流过程的每一个基本活动。属性 (attribute) 用来定义活动的控制信息、可选的实现者、执行分配、运行时的相关信息 (例如优先级)、运行时相关数据 (尤其是在 BRP 中和过程模拟中)。另外, 数据访问限制、转移判断都可以进行定义。必有属性 (Mandatory attributes) 用来说明活动的标识和类型; 只有很少的一部分属性是可选择使用的, 但是在所有的活动类型中都是有公共用法的。其他属性的用法是由活动的类型来决定的, 在下表中给出。

活动的标识和名称在其所在的工作流过程中有效。

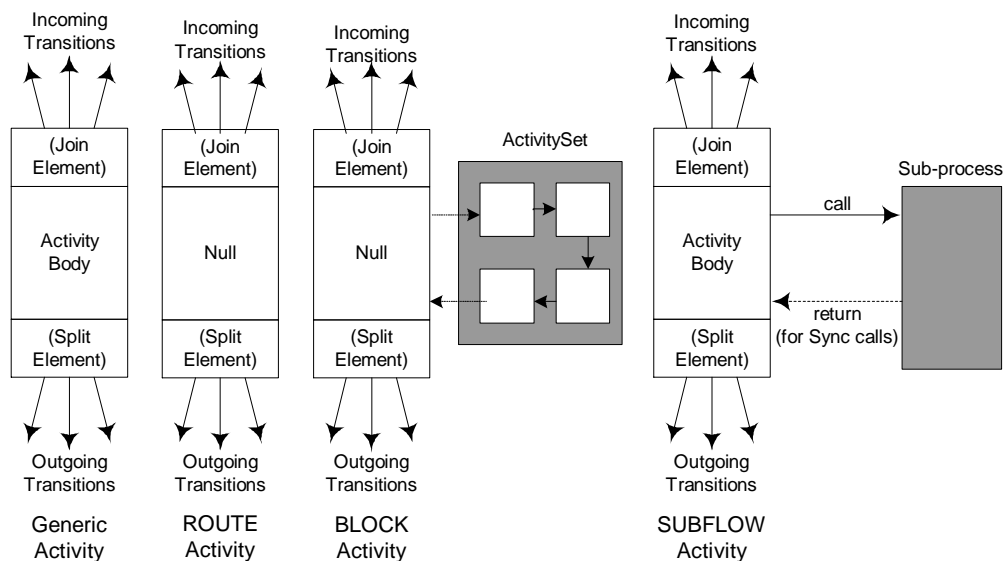
活动描述 (activity description) 用来说明几个不同的活动类型。所有的活动类型都有相同的通用活动属性, 但是其他属性的用法, 尤其是“工作流参与者”、“应用程序分配”和“工作流相关数据”都要由活动的类型来决定。下表给出了不同活动类型的属性 / 实体的用法:

实体类型 (usage within Activity Type)	活动类型				
	实现类型 (Implementation Type)			路径 (Route)	块活动 (BlockActivity)
	无	(应用程序) Application	子流程 (Subflow)		
转移约束 (Transition Restriction)	正常	正常	正常, 并且子流程在 活动内调用/返回	正常, 任何附加控制都 在 Route 活动内实现	正常, 参考相同上下文 中的活动, 但不参考 ActivitySet 中的活动

实体类型 (usage within Activity Type)	活动类型				
	实现类型 (Implementation Type)			路径 (Route)	块活动 (BlockActivity)
	无	(应用程序) Application	子流程 (Subflow)		
参与者分配 (Participant Assignment)	正常	正常	N/A	N/A	N/A
应用程序分配 (Application Assignment)	无	有	N/A	N/A	N/A
工作流相关数据 使用 (Use of workflow Relevant Data)	正常	正常	可能在参数传递中使 用	可能在路径控制条件中 使用	可能在路径控制条件中 用到

使用注意:

转移约束、子流程、路径活动都在转移部分进行说明。通常，正常的转移约束可能在其所在的过程中的活动边界级上说明，然而具体的流程条件（子流程或者路径活动的里面部分），都从活动的内部移到了活动（但是可能要参考其所在过程定义中的活动）中来定义。下图对一般活动和以上活动的不同进行了解释：



当实现类型是“无”时，需要手工来控制工作流活动，并且其实现必须使用明确的信号通知工作流管理系统。这样的活动构成典型的指令，用来指挥工作流参与者承担一个非自动分配的任务，并通知监督者

何时完成。

workflow 相关数据可以被任何活动引用，尽管 workflow 相关数据是用于手工活动中。活动类型是子流程时，任何传递给调用（子）过程的输入参数，都必须在调用过程/活动定义中被声明为 workflow 相关数据，或者从其所在的包中继承。（返回给调用过程的输出参数也有同样的要求）。路径活动和块活动不能直接操作 workflow 相关数据，但是可以在进入/转出控制逻辑的条件表达式中引用相关数据。

```
<xsd:element name="Activity">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="xpdl:Description" minOccurs="0"/>
      <xsd:element ref="xpdl:Limit" minOccurs="0"/>
      <xsd:choice>
        <xsd:element ref="xpdl:Route"/>
        <xsd:element ref="xpdl:Implementation"/>
        <xsd:element ref="xpdl:BlockActivity"/>
      </xsd:choice>
      <xsd:element ref="xpdl:Performer" minOccurs="0"/>
      <xsd:element ref="xpdl:StartMode" minOccurs="0"/>
      <xsd:element ref="xpdl:FinishMode" minOccurs="0"/>
      <xsd:element ref="xpdl:Priority" minOccurs="0"/>
      <xsd:element ref="xpdl:Deadline" minOccurs="0"
maxOccurs="unbounded"/>
      <xsd:element ref="xpdl:SimulationInformation"
minOccurs="0"/>
      <xsd:element ref="xpdl:Icon" minOccurs="0"/>
      <xsd:element ref="xpdl:Documentation" minOccurs="0"/>
      <xsd:element ref="xpdl:TransitionRestrictions"
minOccurs="0"/>
      <xsd:element ref="xpdl:ExtendedAttributes" minOccurs="0"/>
    </xsd:sequence>
    <xsd:attribute name="Id" type="xsd:NMTOKEN" use="required"/>
    <xsd:attribute name="Name" type="xsd:string"/>
  </xsd:complexType>
</xsd:element>
```

```

<xsd:element name="Activities">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="xpdl:Activity" minOccurs="0"
maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:element name="Performer" type="xsd:string"/>

<xsd:element name="Icon" type="xsd:string"/>

```

	描 述
BlockActivity	执行一个活动集 (ActivitySet) 的活动
Deadline	说明最终期限, 如果到达了最终期限活动必须运行
Description	活达的简单描述
Documentation	帮助文件或者活动的描述文件的地址 (例如, 路径和文件名)
Extended Attributes	可选的、开发者自定义的属性, 用来满足特殊需求
Finish Mode	说明在活动完成时, 系统如何运转
Icon	代表活动的图标地址 (路径和文件名)
Id	工作流过程活动 ID
Implementation	“规则”活动, 除路径活动外, 其他的活动必须有此属性。可替代的实现其值用 ‘NO’ 或者 “subflow”
Limit	期望的过程持续时间, 从过程开始计算。达到限制时间 ( Limit time) 的后果在本篇文章中没有说明
Name	工作流过程活动名称
Performer	与工作流参与者相关联, 也可能是一个表达式 默认值: 任何工作流参与者



	描 述
Priority	当活动开始运行时的初始优先级。如果活动中的没有使用此属性，那么将使用过程定义中所定义的优先级 优先级用自然数来标记，从 0 开始，数越大优先级越高。(例如.: 0, 1, ..., n)
Route	虚活动
Simulation Information	活动模拟信息，没有默认值
Start Mode	说明如何运行一个被触发的活动
Transition Restrictions	提供更多的约束和转移描述的上下文相关语义

### 7.5.1. 路径活动 (Route Activity)

路径活动就是虚活动，能够描述层叠转移条件（例如，“IF condition-1 THEN TO Activity-1 ELSE IF condition-2 THEN TO Activity-2 ELSE Activity-3 ENDIF”）。有一些开发者会直接实现层叠转移条件，而不需要路径活动协助。但有一些开发者需要路径活动来协助实现层叠转移条件。开发者和过程定义者都可以设计层叠条件的结构，例如转出活动的出口作成 XOR。有时，用一个活动的转出转移列表，或者一个活动的转入转移列表都无法表示一个转移联合（transition combinations）。在这种情况下，就需要使用一个或多个虚活动。例如：

- 活动的转出转移的离开动作的 XOR 和 AND 联合。
- 活动的进入转移的进入动作的 XOR 和 AND 联合。
- 包含条件与 (AND) 的转移加入到线程的子集，并延续单独线程。

路径活动（虚活动）即没有执行者，也没有应用程序。并且路径活动的运行对 workflow 相关数据，或者应用程序数据没有任何影响。

为了对过程进行模拟，下边的模拟数据值需要被假定为：Duration 0, Cost “0”, WorkingTime 0, WaitingTime 0。 优先级和实例个数的最大值也应该假定。

```
<xsd:element name="Route">  
  <xsd:complexType/>  
</xsd:element>
```

### 7.5.2. 块活动 (Block Activity)

块活动负责执行一个活动集 (ActivitySet)，或者自包含的活动 / 转移映射。从块活动开始执行集合

中的第一个活动，系统一直运行集合中的活动，直到运行到退出活动（没有转出转移的活动）。然后，系统将根据块活动的转出转移的指向推进。

### 7.5.3. 运行控制属性（Execution Control Attributes）

活动中有一些属性是用来说明活动运行控制的详细特性的。自动控制模型明确了，当活动被出发或者停止时的自动化程度。有两个自动控制模型，如下：

- 自动控制模型（Automatic mode）是一个完全由工作流机自动控制的模型，例如一旦满足转移条件，工作流机就会自动执行工作流过程中的活动。同样，当一个活动完成时，自动运行其离开动作。

- 人工模型（Manual mode）需要用户交互来引发活动的开始或结束。在这样的系统中，活动的开始或者完成都是用户交互的结果。

自动控制模型中活动的开始和结束，可以单独定义。

```
<xsd:element name="StartMode">
  <xsd:complexType>
    <xsd:choice>
      <xsd:element ref="xpdl:Automatic" />
      <xsd:element ref="xpdl:Manual" />
    </xsd:choice>
  </xsd:complexType>
</xsd:element>

<xsd:element name="FinishMode">
  <xsd:complexType>
    <xsd:choice>
      <xsd:element ref="xpdl:Automatic" />
      <xsd:element ref="xpdl:Manual" />
    </xsd:choice>
  </xsd:complexType>
</xsd:element>

<xsd:element name="Automatic">
  <xsd:complexType/>
```

```

</xsd:element>

<xsd:element name="Manual">
  <xsd:complexType/>
</xsd:element>

```

	描 述
Start Mode	说明一个活动被出发时如何运行
Automatic	由系统自动触发，此为缺省值
Manual	由用户触发
Finish Mode	活动结束时如何系统如何运行
Automatic	当被调用应用程序完成时自动返回，缺省值
Manual	需要用户来终止一个活动

#### 7.5.4. 活动替代执行方式 (Implementation Alternatives)

假定，活动是自动执行的，其数据是由工作流机控制的。从而，当系统崩溃、异常中断或者活动的取消发生时，工作流相关数据和工作流控制数据都可以恢复（自动，或者使用其他方法），或者进行适当的补偿（不需要为核查数据进行恢复或者补偿）。本版规范中没有包含详细的数据同步控制和数据恢复说明。

```

<xsd:element name="Implementation">
  <xsd:complexType>
    <xsd:choice>
      <xsd:element ref="xpdl:No" />
      <xsd:element ref="xpdl:Tool" maxOccurs="unbounded" />
      <xsd:element ref="xpdl:SubFlow" />
    </xsd:choice>
  </xsd:complexType>
</xsd:element>

```

	描 述
--	-----

	描 述
No implementation	用人工过程来执行活动（例如工作流系统不支持的活动）
Tool	有一个或多个应用程序来执行活动
Subflow	由其他过程执行活动

#### 7.5.4.1. *No Implementation*

**No Implementation** 是说，工作流不能自动的调用应用程序或者方法来执行活动。替代方法已经给出：手工活动（Manual Activity），手工活动需要 **FinishMode** 的值；另一种方式是 隐藏活动（implicit activity），根据处理的需要，只有工作流机知道隐藏活动。例如，工作流中的前过程和后过程活动，在过程开始和结束时都会产生和清除隐藏数据。在这种情况下，**StartMode** 和 **FinishMode** 的值一般为 **Automatic**。（注意，应用程序的初始化，可能仍然由手工活动中的本地控制下的参与者直接完成。这超出了本规范的探讨范围）

```
<xsd:element name="No">
  <xsd:complexType/>
</xsd:element>
```

#### 7.5.4.2. *Tool*

由一个或多个工具来执行活动。工具可以是应用程序，可以通过 **IF3**（接口 3）来调用应用程序。

```
<xsd:element name="Tool">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="xpdl:ActualParameters" minOccurs="0"/>
      <xsd:element ref="xpdl:Description" minOccurs="0"/>
      <xsd:element ref="xpdl:ExtendedAttributes" minOccurs="0"/>
    </xsd:sequence>
    <xsd:attribute name="Id" type="xsd:NMTOKEN" use="required"/>
    <xsd:attribute name="Type">
      <xsd:simpleType>
```

```

        <xsd:restriction base="xsd:NMTOKEN">
            <xsd:enumeration value="APPLICATION"/>
            <xsd:enumeration value="PROCEDURE"/>
        </xsd:restriction>
    </xsd:simpleType>
</xsd:attribute>
</xsd:complexType>
</xsd:element>

```

	描 述				
Actual Parameters	传递给子流程的参数列表.				
Description	工具简短描述				
Extended Attributes	可选的、由开发者自定义的属性，用来满足特殊需求				
Id	应用程序或者方法的 ID				
Type	<table border="0"> <tr> <td>APPLICATION</td> <td>A tool identifier</td> </tr> <tr> <td>PROCEDURE</td> <td>A procedure identifier</td> </tr> </table>	APPLICATION	A tool identifier	PROCEDURE	A procedure identifier
APPLICATION	A tool identifier				
PROCEDURE	A procedure identifier				

#### 7.5.4.3. *subflow*

活动被细化为子流程。子流程可以同步或异步执行。子流程的标识符一般是从其所在的包声明中继承而来。

在异步执行的情况下，活动的执行持续到参考过程定义的过程实例初始化后。在这种情况下，子流程初始化后，执行推进到所有活动后的离开逻辑（离开动作）。并且被调用过程不返回参数。同步执行的子流程，需要使用其他方法（例如事件）来处理。在本文档中不进行说明。这种子流程运行方式特性化为链式子流程运行（*chained subflow operation*）。

在同步执行的情况下，在过程实例被初始化后，活动将被挂起。在这个过程实例运行终止后，活动被恢复。在子流程完成时，在调用与被调用过程间可能要使用返回参数。这种形式的子流程被特性化为等级子流程运行（*hierarchic subflow operation*）。

```

<xsd:element name="SubFlow">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="xpdl:ActualParameters" minOccurs="0"/>

```

```

</xsd:sequence>

<xsd:attribute name="Id" type="xsd:string" use="required"/>

<xsd:attribute name="Execution">

  <xsd:simpleType>

    <xsd:restriction base="xsd:NMTOKEN">

      <xsd:enumeration value="ASYNCHR"/>

      <xsd:enumeration value="SYNCHR"/>

    </xsd:restriction>

  </xsd:simpleType>

</xsd:attribute>

</xsd:complexType>

</xsd:element>

```

	描 述
Actual Parameters	传递给子流程的参数列表
Execution	ASYNCHR 异步执行
	SYNCHR 同步执行
Id	所调用的工作流过程 ID

### 7.5.5. 活动与执行者间关系 (Performer Relationship)

活动和执行者间的关系，由参与者分配 (Participant Assignment) 属性给出。提供了一个活动与工作流参与者实体之间的连接。缺省值：所有参与者。

Performer 属性中的工作流参与者标识符必须在其所在的工作流过程定义中声明，或者从其所在的包中继承而来。

问题是：表达式赋值是否会导致执行者集合为空；或者在运行时期，是否一个不唯一执行者要被工作流管理系统处理，或者在定义的地方指明，由外部资源库来处理或者由组织模型处理。运行时期的解决办法已经超出了本文的范围。

- 在第一种情况下（空集），工作流机可能会不停的重试，或者工作流机发信号给过程监督者。经常使用的方法属于 WFMS 范畴，不是本文所要讲述的部分。

- 第二种情况下（不唯一）发生的地方，可能是执行者定义为 function/skill 类型或者一个组织单元，从而其自身是一个参与者集合的容器。在这些情况下，参与者分配所采用的方法属于 WFMS 范畴，不是本文要讲述的部分。通常的情节如下：

a、当一个活动包含多个可以相似执行的工作项目时，每个单独的工作项目可能会预分配给一系列的执行者。

b、在其他情况下，会根据局部负荷平衡算法来为活动分配执行者，或者把活动预分配给，在其 workflow 列表中的多个可能的执行者，并且最后分配给第一个可以使用的参与者。（要由 workflow 机来提供适当的操作）

c、把一个活动分配给一个组织机构（例如，一个部门），可能会导致这个活动提供给组织机构中的所有成员，并且分配给第一个可以得到的参与者，或者组织机构的管理者对活动的分配情况进行重新定向。

在所有的情况中，元模型中定义的参与者分配，以及用 XPD L 表示的参与者分配，只是把活动与定义的参与者联系起来，并且无论定义的参与者是自动或者非自动的情况下，都是一样的。处理那些情况的 workflow 机的局部行为和资源库，或者组织模型都是没有定义过的。

### 7.5.6. 最终期限 (Deadline)

```
<xsd:element name="Deadline">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="DeadlineCondition" minOccurs="1"
maxOccurs="1"/>
      <xsd:element name="ExceptionName" minOccurs="1"
maxOccurs="1"/>
    </xsd:sequence>
    <xsd:attribute name="Execution">
      <xsd:simpleType>
        <xsd:restriction base="xsd:NMTOKEN">
          <xsd:enumeration value="ASYNCHR"/>
          <xsd:enumeration value="SYNCHR"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:attribute>
  </xsd:complexType>
</xsd:element>
```

	描 述
Execution	<p>如果达到（超出）最终期限，系统如何运行</p> <p>ASYNCHR                      异步产生最终期限。暗含一个与离开（AND SPLIT）操作，在此处活动继续运行，并且产生另外一个线程，按照异常转移（exception transition）的路径推进。活动还可能遇到另外一个最终期限，因为其是一直运行的。</p> <p>SYNCHR                        非法完成的活动，按照异常转移（exception transition）</p>
DeadlineCondition	<p>指明最终期限的表达式。表达式的计算至少依靠或者包含以下几方面：</p> <p>活动的相对开始时间 (2 days)</p> <p>确定时间 (January 1) or (January 1, 2002)</p> <p>由 workflow 相关数据计算出来的时间 (varName days)</p>
ExceptionName	<p>达到最终期限时将产生的异常的名称</p>

如果一个活动运行时间达到了一个固定时期，使用最终期限用来产生一个异常。

在达到最终期限时，会产生一个异常条件，流程会按照适当的异常转移向前推进。如果最终期限是同步执行的，活动在按照异常转移路径推进前，会被终止。如果最终期限是异步执行的，会执行其隐含的与离开（AND SPLIT），并产生一个新的线程按照适当的异常转移向前推进。异步异常会产生侧面影响，需谨慎使用。一些典型的侧面影响在本部分后面讨论。

下边是一个最终期限的例子。例子中，一个异步异常“notifyException”将在3天后产生，在此活动还将正常运行。如果活动继续运行5天，那么其将被终止，并产生一个“timeoutException”。

例子：

```

<Deadline Execution="ASYNCHR">
    <DeadlineCondition>3 days</DeadlineCondition>
    <ExceptionName>notifyException</ExceptionName>
</Deadline>
<Deadline Execution="SYNCHR">
    <DeadlineCondition>5 days</DeadlineCondition>
    <ExceptionName>timeoutException</ExceptionName>
</Deadline>

```

DeadlineConditions 可以是相对或者绝对时间，也可以使用 workflow 相关数据。如果同步执行的最终期限在块活动或者子流程中发生，那么终止活动包括终止块中的所有线程、终止子流程和其中的所有线程，同时终止其嵌套子流程。从建模观点来说，这可以看作是“立即终止”。如果不想按照上述方式执行，例



如要结束一个正在处理的手工活动，那么需要对此行为进行描述。

最终期限的异步执行是一个很有用的方式，可以用其进行中间消息传递，也可通过改变 workflow 相关数据来完美的终止一个过程。但是异步异常可能会产生条件竞争，和一些侧面影响。例如，异步异常被处理的同时，在运行的活动可以完成。另外，由于执行了一个隐含的分支（split），这时如果处理异步异常的过程加入到最终期限的处理线程队列中，那么流程控制会变得非常复杂。workflow 设计者必须小心处理条件竞争，避免不期望的侧面影响。

### 7.5.7. 过程模拟信息（Simulation Information）

```
<xsd:element name="SimulationInformation">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="xpdl:Cost" />
      <xsd:element ref="xpdl:TimeEstimation" />
    </xsd:sequence>
    <xsd:attribute name="Instantiation">
      <xsd:simpleType>
        <xsd:restriction base="xsd:NMTOKEN">
          <xsd:enumeration value="ONCE" />
          <xsd:enumeration value="MULTIPLE" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:attribute>
  </xsd:complexType>
</xsd:element>

<xsd:element name="TimeEstimation">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="xpdl:WaitingTime" minOccurs="0" />
      <xsd:element ref="xpdl:WorkingTime" minOccurs="0" />
      <xsd:element ref="xpdl:Duration" minOccurs="0" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

```

        </xsd:complexType>
    </xsd:element>

    <xsd:element name="WaitingTime" type="xsd:string"/>
    <xsd:element name="WorkingTime" type="xsd:string"/>
    <xsd:element name="Duration" type="xsd:string"/>
    <xsd:element name="Cost" type="xsd:string"/>

```

为了达到高吞吐量，Instantiation 属性定义了，一个活动可以被实例的次数。可以是一次活多次。

	描 述
Cost	平均代价
Duration	执行一个任务的预期持续时间
Instantiation	定义一个活动可以被实例的次数：一次或者多次（ONCE or MULTIPLE）  ONCE                      The Activity can only be instantiated once. Default.  MULTIPLE                  The Activity can be instantiated multiple times.
Time Estimation	一个活动的预计持续时间(用持续时间单元来计算) (summary of working time, waiting time, and duration)
Waiting Time	平均等待时间（用持续时间单元来计算）
Working Time	平均工作时间（用持续时间来计算）

### 7.5.8. 转移约束（Transition Restrictions）

```

<xsd:element name="TransitionRestriction">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="xpd1:Join" minOccurs="0"/>
            <xsd:element ref="xpd1:Split" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>

```

```
<xsd:element name="TransitionRestrictions">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="xpdl:TransitionRestriction"
        minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

	描 述
Join	说明活动的进入转移
Split	活动的转出转移是分路的 (SPLIT-ed)

#### 7.5.8.1. *Join*

用 join 可以描述一个活动的多个进入转移。

```
<xsd:element name="Join">
  <xsd:complexType>
    <xsd:attribute name="Type">
      <xsd:simpleType>
        <xsd:restriction base="xsd:NMTOKEN">
          <xsd:enumeration value="AND" />
          <xsd:enumeration value="XOR" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:attribute>
  </xsd:complexType>
</xsd:element>
```

	描 述
Type	AND 同时加入，过程实例中此活动的所有进入转移的并发线程。这需要同步执行，同步运行的线程数量受，转移前的 AND SPLIT (s) 的条件限制

AND Join: 可以看作是活动的“预处理集合点”；直到活动的所有进入转移的条件都满足，活动才被初始化。

XOR Join : 只要活动的任一进入转移的转移条件得到满足，活动就被初始化。

### 7.5.8.2. Split

用 Split 可以描述一个活动的多个转出转移。

```
<xsd:element name="Split">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="xpd:TransitionRefs" minOccurs="0" />
    </xsd:sequence>
    <xsd:attribute name="Type">
      <xsd:simpleType>
        <xsd:restriction base="xsd:NMTOKEN">
          <xsd:enumeration value="AND" />
          <xsd:enumeration value="XOR" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:attribute>
  </xsd:complexType>
</xsd:element>

<xsd:element name="TransitionRef">
  <xsd:complexType>
    <xsd:attribute name="Id" type="xsd:NMTOKEN" use="required" />
  </xsd:complexType>
</xsd:element>
```

```

        </xsd:complexType>

</xsd:element>

<xsd:element name="TransitionRefs">

    <xsd:complexType>

        <xsd:sequence>

            <xsd:element ref="xpdl:TransitionRef"

                minOccurs="0" maxOccurs="unbounded" />

        </xsd:sequence>

    </xsd:complexType>

</xsd:element>

```

	描 述
Transition Refs	活动的所有转出转移列表。 每个转移都有一个唯一的 ID
Type	<p>AND      定义可能并发执行的、代表这个活动转出转移的线程</p> <p>          如果有条件转移，并发线程的数量由每个转移的转移条件决定，这些转移条件的计算是并发进行的</p> <p>XOR      活动的转出转移列表，其中转移都是根据转移条件来选择执行的。</p> <p>          选择哪条转移路径，需要根据每个转移的转移条件来决定，按照转移在列表中的顺序来对转移条件进行逐个判断（计算），满足则执行，不满足就判断（计算）下一个</p> <p>          如果计算到一个无条件转移，或者一个转移的条件是 OTHERWISE，那么结束对表中转移的判断</p>

有条件转移构成的 AND SPLIT，可以看做“条件与(conditional AND)”、“多重选择 OR(multiple-choice OR)”，或者是“不独一 OR(nonexclusive OR)”。并发线程的实际数量在转移条件判断（计算）时明确。在 AND SPLIT 情况下，过程实例（或者过程实例的线程）会进入到交叉的、一系列独立运行的线程中，这些线程是由转移条件判断后产生的。（注意：这时候不需要转移列表，因为活动的所有转出转移都需要被判断，并与顺序无关）

如果在构成 AND SPLIT 的转移中，有一个转移的转移条件类型是 OTHERWISE，那么要进行两步判断。第一步，除了转移条件类型是 OTHERWISE 的转移外，对其余所有转移的转移条件进行判断。如果转移条件没有满足的（包括无条件转移），则进行第二步，对所有的转移的转移条件进行判断，包括转移条件类型是 OTHERWISE 的转移。（在一个活动的转出转移列表中，只允许有一个转移的转移条件类型是 OTHERWISE）

使用转移条件类型为 OTHERWISE 的转移，可以保证过程运行状态非常清楚。（例如，一个活动至少

有一个转出转移会被执行)

### 7.5.9. 一致分类 (Conformance Classes)

使用一致分类 (Conformance Classes) 来对活动转移网进行约束。

在包中定义了以边几个一致分类:

- NON-BLOCKED

无限制

- LOOP-BLOCKED

过程定义中的活动和转移构成一个非循环图

- FULL-BLOCKED

每个 JOIN (或者 SPLIT) 都有一个明确的、相应的、同类型的 SPLIT (或者 JOIN) 与之对应。在 AND SPLIT 中可以存在无条件转移; 在 XOR SPLIT 中, 如果存在有条件转移, 那么就需要一个无条件转移, 或者转移条件的类型是 OTHERWISE 的转移。(例如, 不允许存在不确定的转移判断)

## 7.6. 转移信息 (Transition Information)

转移信息, 描述活动间的转移, 也描述在工作流运行期间转移发生与否的条件。更进一步的控制和构造约束, 在活动定义中进行表达。

一个过程定义, 可以看作是由活动点间的边构成的网络 (例如, 工作流过程图)。边都是有向的, 而且由一对活动来决定 (源活动点、目标活动点)。

活动网的边, 可能会被标记上转移条件 (Transition conditions)。标有转移条件的边, 当条件满足时发生转移。如果没有标记转移条件, 那么就当作转移条件永远为真。

如果一个活动有多个进入或者转出转移, 更多的用来表示控制约束和条件判断的选项, 在活动实体定义中给出。(AND/XOR 与 SPLIT/JOIN 的不同组合)

用一个指向转移路径起点活动的转移, 能表示一个循环。典型的, 在循环路径上, 至少有一个活动有多个转出转移, 其中的一个或多个转移包含有退出循环的条件。

转移信息中的标识符、名称等在其所在的工作流定义中有效。

可以定义或者同步多个控制线程, 也可以把活动间的转移序列化。

```
<xsd:element name="Transition">  
    <xsd:complexType>
```

```

        <xsd:sequence>
            <xsd:element ref="xpd1:Condition" minOccurs="0"/>
            <xsd:element ref="xpd1:Description" minOccurs="0"/>
            <xsd:element ref="xpd1:ExtendedAttributes" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attribute name="Id" type="xsd:NMTOKEN" use="required"/>
        <xsd:attribute name="From" type="xsd:NMTOKEN" use="required"/>
        <xsd:attribute name="To" type="xsd:NMTOKEN" use="required"/>
        <xsd:attribute name="Name" type="xsd:string"/>
    </xsd:complexType>
</xsd:element>

<xsd:element name="Transitions">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="xpd1:Transition" minOccurs="0"
maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>

```

	描述
Condition	转移条件表达式，以 workflow 相关数据为基础。 (例如: 'Contract' = 'SMALL' OR 'Contract' <\$20,000). 缺省值: TRUE
Description	转移的简短文本描述
Extended Attributes	可选的、用户自定义的属性，用来满足特殊需求
From	转移的源活动 (活动标识符)
Id	转移的 ID
Name	转移名称
To	转移的目标活动 (活动标识符)

### 7.6.1. Condition

```
<xsd:element name="Condition">
  <xsd:complexType mixed="true">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
      <xsd:element ref="xpdl:Xpression"/>
    </xsd:choice>
    <xsd:attribute name="Type">
      <xsd:simpleType>
        <xsd:restriction base="xsd:NMTOKEN">
          <xsd:enumeration value="CONDITION"/>
          <xsd:enumeration value="OTHERWISE"/>
          <xsd:enumeration value="EXCEPTION"/>
          <xsd:enumeration value="DEFAULTEXCEPTION"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:attribute>
  </xsd:complexType>
</xsd:element>

<xsd:element name="Xpression">
  <xsd:complexType mixed="true">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
      <xsd:any minOccurs="0" maxOccurs="unbounded"/>
    </xsd:choice>
  </xsd:complexType>
</xsd:element>
```



	描 述
Type	<p>定义转移条件的类型，有效值如下：</p> <p>CONDITION                    转移条件得到满足，转移才会被执行</p> <p>OTHERWISE                    说明此转移为缺省转移，如果其他转移的转移条件都不能得到满足，则执行此转移</p> <p>EXCEPTION                    异常转移，如果产生了一个异常，并且转移条件得到满足，则执行该转移</p> <p>DEFAULTEXCEPTION            异常缺省转移，如果产生了异常，并且其他异常转移的转移条件都得不到满足，则执行此转移</p>
Xpression	用 XML 标签描述的条件表达式

### 7.6.1.1. 异常转移条件 (Exception Conditions)

使用 EXCEPTION 和 DEFAULTEXCEPTION 类型的转移，可以指定当产生异常时，被执行的转移分支。EXCEPTION 类型与 CONDITION 类型相同，DEFAULTEXCEPTION 与 OTHERWISE 类型相对应。转移条件中可能包含异常的名字，或者更复杂的表达式。除最终期限外，异常都在特殊工作流机中以某种特殊形式产生。象规则转移一样，只有源活动结束后，异常转移才能发生。

下边的举例说明，从一个活动引出的所有转移，包括异常转移。分支 1 和 2 在正常情况下被处理；分支 3 和 4 在产生异常时被处理：

```
<Transitions>
  <Transition Id="branch1" From="CheckBalance" To="ProcessRequest">
    <Condition Type="CONDITION">Balance > 1000</Condition>
  </Transition>
  <Transition Id="branch2" From="CheckBalance" To="InsufFunds">
    <Condition Type="OTHERWISE" />
  </Transition>
  <Transition Id="branch3" From="CheckBalance" To="MessageDisplay">
    <Condition Type="EXCEPTION">ATMDownException</Condition>
  </Transition>
  <Transition Id="branch4" From="CheckBalance" To="SendAlarm">
    <Condition Type="DEFAULTEXCEPTION" />
  </Transition>
</Transitions>
```

```
</Transitions>
```

## 7.7. workflow参与者 (Workflow Participants)

workflow参与者是下边类型中的一个：资源集、资源、组织机构、角色、人、或者系统。资源和角色是抽象参与者。workflow参与者定义，是活动与其真正执行者间的抽象层。在运行时期，这些抽象定义被计算，并且分配给真正的人类或者程序。

workflow参与者实体声明的标识符的有效范围是其所在的工作流定义或者过程模型定义。

外部资源库或者组织模型可能会包含一些附加信息，用来对基本的参与者类型进行补充。

```
<xsd:element name="Participant">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="xpdl:ParticipantType"/>
      <xsd:element ref="xpdl:Description" minOccurs="0"/>
      <xsd:element ref="xpdl:ExternalReference" minOccurs="0"/>
      <xsd:element ref="xpdl:ExtendedAttributes" minOccurs="0"/>
    </xsd:sequence>
    <xsd:attribute name="Id" type="xsd:NMTOKEN" use="required"/>
    <xsd:attribute name="Name" type="xsd:string"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="Participants">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="xpdl:Participant" minOccurs="0"
maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

	描 述
Description	workflow 参与者描述
ExternalReference	外部参与者规范参考
Extended Attributes	可选的、用户自定义的属性，用来满足特殊需求
Id	workflow 参与者定义 ID
Name	执行者名称
Participant Type	workflow 参与者实体类型

### 7.7.1. workflow 参与者实体类型

workflow 参与者实体类型属性，指明参与者是个人、组织机构或者是一个抽象的资源，例如机器。

```
<xsd:element name="ParticipantType">
  <xsd:complexType>
    <xsd:attribute name="Type" use="required">
      <xsd:simpleType>
        <xsd:restriction base="xsd:NMTOKEN">
          <xsd:enumeration value="RESOURCE_SET" />
          <xsd:enumeration value="RESOURCE" />
          <xsd:enumeration value="ROLE" />
          <xsd:enumeration value="ORGANIZATIONAL_UNIT" />
          <xsd:enumeration value="HUMAN" />
          <xsd:enumeration value="SYSTEM" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:attribute>
  </xsd:complexType>
</xsd:element>
```

	描	述
Type	RESOURCE_SET	资源集
	RESOURCE	一个明确的资源
	ROLE	可以通过角色或者技能集来选择执行者。在这里“角色”在指一个人在的职能，作为职能讲所以角色必须是唯一的，可能会定义与角色相协作的人员，或者是与其相关的人员列表
	ORGANIZATIONAL_UNIT	组织模型中的部门，或者其他单位
	HUMAN	通过应用程序与系统交互的人，应用程序为参与者提供用户接口
	SYSTEM	自动运行

## 7.8. workflow 相关数据 (Workflow Relevant Data)

workflow 相关数据描述 workflow 过程定义或者包定义中的变量。典型的用于维护判断数据（转移条件需要用到）或者参考数据值（参数），这些数据在活动或者子流程间传递。相关数据与 workflow 参与者数据是不同的，参与者数据的数据管理或者访问，完全由被调用应用程序负责，并且 workflow 管理系统不能访问参与者数据。workflow 相关数据表定义了 workflow 过程所需的所有数据对象。Data Type 属性，明确指定 workflow 管理系统定义适当的进行数据存储的数据对象，所需要的所有信息，这些信息将由激活的 workflow 过程实例所处理。

可以在 workflow 过程或者包中对 workflow 相关数据进行定义。因为 workflow 过程只能被过程定义中的实体所访问，而包可以用来对过程实体的参数进行定义，所以两者中所定义的 workflow 相关数据的有效范围是不同的。

当参数传递给现行模型定义以外的被调用子流程时（例如，远程过程调用），过程设计者要确保在各个参数集合中的数据类型的兼容性。

```
<xsd:element name="DataField">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="xpdl:DataType" />
      <xsd:element ref="xpdl:InitialValue" minOccurs="0" />
      <xsd:element ref="xpdl:Length" minOccurs="0" />
      <xsd:element ref="xpdl:Description" minOccurs="0" />
      <xsd:element ref="xpdl:ExtendedAttributes" minOccurs="0" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

```
</xsd:sequence>

<xsd:attribute name="Id" type="xsd:NMTOKEN" use="required"/>

<xsd:attribute name="Name" type="xsd:string"/>

<xsd:attribute name="IsArray" default="FALSE">

  <xsd:simpleType>

    <xsd:restriction base="xsd:NMTOKEN">

      <xsd:enumeration value="TRUE"/>

      <xsd:enumeration value="FALSE"/>

    </xsd:restriction>

  </xsd:simpleType>

</xsd:attribute>

</xsd:complexType>

</xsd:element>

<xsd:element name="DataFields">

  <xsd:complexType>

    <xsd:sequence>

      <xsd:element ref="xpd1:DataField" minOccurs="0"
maxOccurs="unbounded"/>

    </xsd:sequence>

  </xsd:complexType>

</xsd:element>

<xsd:element name="DataType">

  <xsd:complexType>

    <xsd:group ref="xpd1:DataTypes"/>

  </xsd:complexType>

</xsd:element>

<xsd:element name="InitialValue" type="xsd:string"/>

<xsd:element name="Length" type="xsd:string"/>
```

	描 述
Data Type	过程变量的数据类型
Description	数据定义的简短描述
Extended Attributes	可选的、用户自定义的属性，用来满足特殊需求
Id	工作流相关数据的 ID
Initial Value	为运行时期进行的数据预分配
Is Array	是否为数组
Length	数据的长度
Name	工作流相关数据的名称

## 7.9. 数据类型 (Data Type)

数据类型由标准的类型集构成，这些标准的类型可以用来对工作流相关数据、形式参数、工作流过程进行数据描述。可以在 `TypeDeclaration` 中声明一个新的数据类型，并能在所有可以使用标准数据类型的地方使用这个新数据类型。可以从下边的类型集合中选择一个数据类型：

```
<xsd:group name="DataTypes">
  <xsd:choice>
    <xsd:element ref="xpdl:BasicType"/>
    <xsd:element ref="xpdl:DeclaredType"/>
    <xsd:element ref="xpdl:SchemaType"/>
    <xsd:element ref="xpdl:ExternalReference"/>
    <xsd:element ref="xpdl:RecordType"/>
    <xsd:element ref="xpdl:UnionType"/>
    <xsd:element ref="xpdl:EnumerationType"/>
    <xsd:element ref="xpdl:ArrayType"/>
    <xsd:element ref="xpdl:ListType"/>
  </xsd:choice>
</xsd:group>
```

	描 述
Array Type	有固定大小的数据集，其中的数据均为同种类型 (反对使用).
Basic Type	简单类型: STRING, INTEGER, FLOAT, DATETIME, REFERENCE, BOOLEAN, or PERFORMER.
Declared Type	在 TypeDeclaration 中声明的数据类型
Enumeration Type	变量或者参数的合法值集合 (反对使用).
ExternalReference	在外部文档中的类型定义参考
List Type	没有边界的数据集，其中的数据均为同种类型 (反对使用).
Record Type	由多个成员构成的集合，成员的类型可以不同 (反对使用).
SchemaType	使用 XML schema 定义的数据类型
Union Type	由多个成员构成的集合，并且只有一个成员会被数据的实例所使用 (反对使用).

### 7.9.1. 基本数据类型(Basic Data Type)

```
<xsd:element name="BasicType">
  <xsd:complexType>
    <xsd:attribute name="Type" use="required">
      <xsd:simpleType>
        <xsd:restriction base="xsd:NMTOKEN">
          <xsd:enumeration value="STRING"/>
          <xsd:enumeration value="FLOAT"/>
          <xsd:enumeration value="INTEGER"/>
          <xsd:enumeration value="REFERENCE"/>
          <xsd:enumeration value="DATETIME"/>
          <xsd:enumeration value="BOOLEAN"/>
          <xsd:enumeration value="PERFORMER"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:attribute>
  </xsd:complexType>
</xsd:element>
```

	描	述
Type	STRING	有限长度的字符序列
	FLOAT	浮点型或者双精度数字，最大值在 XPDL 中没有定义
	INTEGER	整型
	REFERENCE	参考外部数据类型—现已反对使用。推荐使用 ExternalReference 来引用一个外部数据类型
	DATETIME	一个准确的时间实例。日期格式在 XPDL 中没有说明
	BOOLEAN	布尔型数据类型，取值为 True 或者 False。True 和 False 的内部表现形式，在 XPDL 中没有说明
	PERFORMER	执行者类型，其值为声明了的工作流参与者

## 7.9.2. 复杂数据类型（Complex Data Type）

XPDL 中可以进行复杂数据类型定义，例如数组（arrays）、记录（records）、联合（unions）、和列表（lists）。复杂数据类型使用 SchemaType 进行定义。过去使用 RecordType, UnionType, EnumerationType, ArrayType, and ListType 来定义复杂数据类型，现在反对使用。但在 XPDL 中还保留了他们，为的是与以前的版本兼容。

### 7.9.2.1. Schema Type

Schema Type 允许开发者使用 XML schema 语法进行数据类型定义。可以使用 Schema Type 来定义一个符合 schema 的 XML 字符串。

```

<xsd:element name="SchemaType">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:any namespace="##other" minOccurs="0"
maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

```



下边的例子，可以描述 C++或者 JAVA CLASS、C 结构体、XML 字符串：

```
<SchemaType>
  <schema xmlns="http://www.w3.org/2000/10/XMLSchema">
    <element name="PO">
      <complexType>
        <sequence>
          <element name="CustomerName" type="string"/>
          <element name="Address" type="string"/>
          <element name="OrderNumber" type="string"/>
        </sequence>
      </complexType>
    </element>
  </schema>
</SchemaType>
```

#### 7.9.2.2. 记录类型 (Record Type)

```
<xsd:element name="RecordType">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="xpdl:Member" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:element name="Member">
  <xsd:complexType>
    <xsd:group ref="xpdl:DataTypes"/>
  </xsd:complexType>
</xsd:element>
```

	描 述
Member	记录中的一个字段
DataTypes	Member 的数据类型

### 7.9.2.3. 联合类型 (Union Type)

```

<xsd:element name="UnionType">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="xpdl:Member" maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:element name="Member">
  <xsd:complexType>
    <xsd:group ref="xpdl:DataTypes" />
  </xsd:complexType>
</xsd:element>

```

	描 述
Member	联合中的字段 (成员)
DataTypes	Member 的数类型

### 7.9.2.4. 枚举类型 (Enumeration Type)

```

<xsd:element name="EnumerationType">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="xpdl:EnumerationValue"
maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

```

```

        </xsd:sequence>

    </xsd:complexType>

</xsd:element>

<xsd:element name="EnumerationValue">
    <xsd:complexType>
        <xsd:attribute name="Name" type="xsd:NMTOKEN" use="required"/>
    </xsd:complexType>
</xsd:element>

```

	描 述
Enumeration Value	元素，用来定义枚举中的一个值
Name	值的名称

#### 7.9.2.5. 数组类型 (Array Type)

```

<xsd:element name="ArrayType">
    <xsd:complexType>
        <xsd:group ref="xpd1:DataTypes">
        </xsd:group>

        <xsd:attribute name="LowerIndex" type="xsd:NMTOKEN"
use="required"/>

        <xsd:attribute name="UpperIndex" type="xsd:NMTOKEN"
use="required"/>

    </xsd:complexType>
</xsd:element>

```

	描 述
DataTypes	数组的数据类型
Lower Index	ArrayType 的下界索引值
Upper Index	ArrayType 的上界

### 7.9.2.6. 列表类型 (List Type)

```
<xsd:element name="ListType">
  <xsd:complexType>
    <xsd:group ref="xpd1:DataTypes">
    </xsd:group>
  </xsd:complexType>
</xsd:element>
```

	描	述
DataTypes	列表中数据的数据类型	

## 7.9.3. 开发者声明的数据类型 (Declared Data Type)

在可以使用标准 XPD L 数据类型的任何地方，都可以重复使用复杂数据类型定义。在 `TypeDeclaration` 中定义数据类型，然后使用 `DeclaredType` 标签引用这个数据类型。

### 7.9.3.1. 类型声明 (Type Declaration)

```
<xsd:element name="TypeDeclaration">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:group ref="xpd1:DataTypes">
      <xsd:element ref="xpd1:Description" minOccurs="0"/>
      <xsd:element ref="xpd1:ExtendedAttributes" minOccurs="0"/>
    </xsd:sequence>
    <xsd:attribute name="Id" type="xsd:ID" use="required"/>
    <xsd:attribute name="Name" type="xsd:string"/>
  </xsd:complexType>
</xsd:element>
```

```

<xsd:element name="TypeDeclarations">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="xpd1:TypeDeclaration"
        minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

```

	描 述
DataTypes	数据类型
Description	数据类型的非正式描述
ExtendedAttributes	可选的、用户自定义的属性，用来满足特殊需求
Id	TypeDeclaration.的 ID
Name	TypeDeclaration.的名称

使用 SchemaType 定义一个可以重复使用的“定货单”类型：

```

<TypeDeclarations>
  <TypeDeclaration Id="POType" Name="PurchaseOrder">
    <SchemaType>
      <schema xmlns="http://www.w3.org/2000/10/XMLSchema">
        <element name="PO">
          <complexType>
            <sequence>
              <element name="CustomerName" type="string"/>
              <element name="Address" type="string"/>
              <element name="OrderNumber" type="string"/>
            </sequence>
          </complexType>
        </element>
      </schema>
    </SchemaType>
  </TypeDeclaration>
</TypeDeclarations>

```

```

        </schema>

        </SchemaType>

    </TypeDeclaration>

</TypeDeclarations>

```

### 7.9.3.2. Declared Type

```

<xsd:element name="DeclaredType">
    <xsd:complexType>
        <xsd:attribute name="Id" type="xsd:IDREF" use="required"/>
    </xsd:complexType>
</xsd:element>

```

	描 述
Id	所使用的数据类型在 TypeDeclaration.的 ID

例如，使用 DeclaredType 可以，定义多个 PType 类型的变量：

```

<DataFields>
    <DataField Id="newPO">
        <DataType>
            <DeclaredType Id="PType"/>
        </DataType>
    </DataField>
    <DataField Id="checkedPO">
        <DataType>
            <DeclaredType Id="PType"/>
        </DataType>
    </DataField>
</DataFields>

```

## 8. 工作流例子

用这个例子来说明 XPD L 的一些特性，并不代表实际的过程或者实现此过程的最好办法。

用一个定单登记系统作例子，定单格式化成为相应的字符串后进入系统。包中有一个主要过程和两个子过程。下边对定单登记过程、将用到的数据类型、扩展属性，还有外部参考，做了一下简单讨论。最后给出 XPD L。

### 8.1. 定单登记过程

#### 8.1.1. EOroder 主过程

主过程把格式化后的字符串作为输入数据，并且显示定单是否符合要求。主过程包含如下步骤：

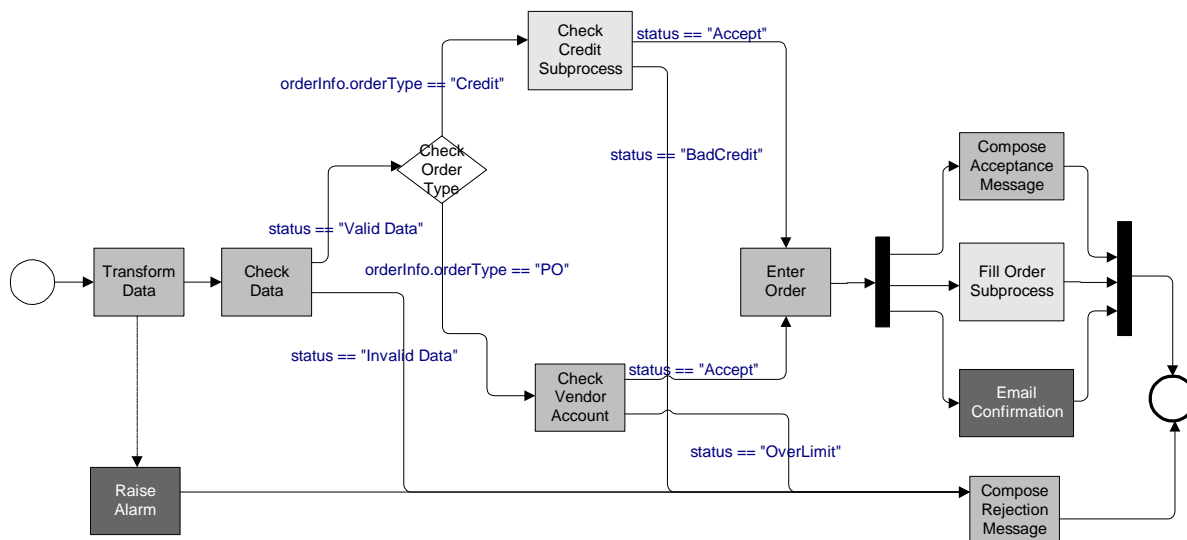
- 先把字符串转变成复杂的数据对象。如果产生异常，发出警告并拒绝对单。
- 检查数据的正确性。
- 判断使用定货单支付，还是信用卡支付。
- 把信用卡定货的定单转给，能使用信用卡定货的子过程处理。
- 使用应用程序处理定货单支付的货单，应用程序检查购货者的记录并且批准购买的数量。
- 货单存入数据库，并为其分配一个货单号。
- 以下三个活动并发执行：

定单接受信息返回给终端用户。

异步调用一个子过程来填写定单。

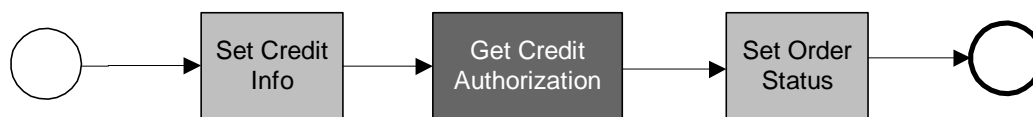
定单确认 email 发送给终端用户。这个活动由工作流管理系统负责。使用 `ExtendedAttributes` 来说明，系统需要 email 的信息。

- 如果定单被拒绝，或者因为其填写不准确，或者由于没有得到批准，把拒绝信息返回给客户。



### 8.1.2. CreditCheck 子过程

CreditCheck 子过程，用输入参数创建 CreditInfo 对象，并把信息发送给信用卡 web service 进行审批。Web server 返回转变成 OrderStatus 类型的状态字符串给调用过程。



### 8.1.3. FillOrder 子过程

FillOrder 子过程负责处理货物发派送和结帐。此过程中包含一个参与者“Shipper”。

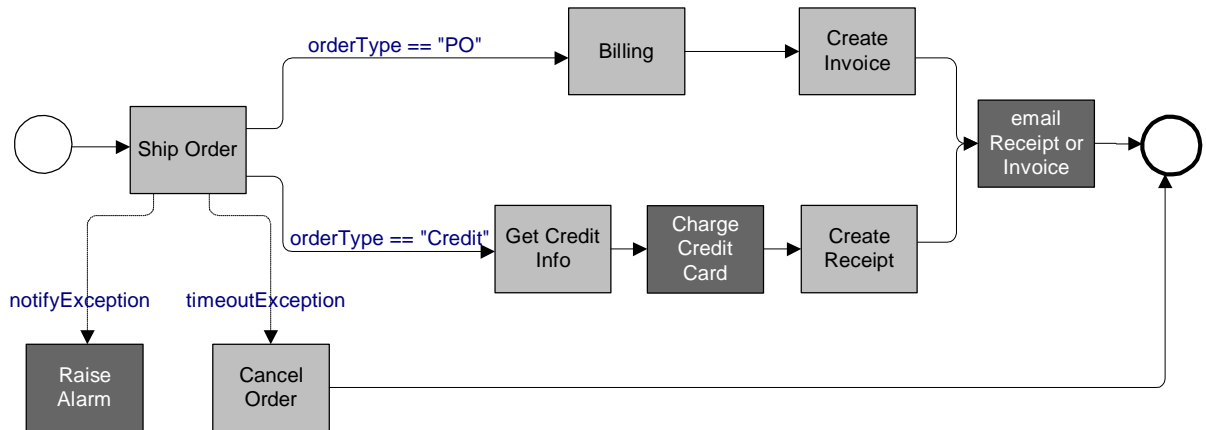
- 第一个活动，显示定单信息给派货人，派货人负责运送定单中的货物，并记录货单中每项货物的状态。应用程序返回定单的状态——是否完成或者退货。这个活动包含有一些最终期限。如果活动在 3 天内没有完成，就会产生 notifyException 异常，并发出警告。如果活动在 5 天内还没有完成，将产生 timeoutException 异常，并切取消定单。

- 然后，判断是 PO (Purchase Order) 还是信用卡定单。
- 如果是 PO，则发送给 billing 系统，随后生成一个电子发票，并存储在服务器中。
- 如果是信用卡定单，则发送给负责收费的信用卡 web service，并且生成一个电子收据，存储在



服务器中。

● 最后，把发票或者收据作为 email 的附件发送给客户。使用 ExtendedAttributes 来说明需要 email 信息。



## 8.2. 类型声明 (Type Declarations)

为工作流定义了一系列的数据类型：

● 用单独的 schema 文档定义一个 Order 数据类型，使用 ExternalReference 引用此数据类型：

```
<TypeDeclaration Id="Order" Name="Order">
    <ExternalReference location="C:\wfmc\sample\orderschema.xsd"/>
</TypeDeclaration>
```

Schema:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified">
    <xsd:element name="Order">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="Items">
                    <xsd:complexType>
```

```
<xsd:sequence>
  <xsd:element name="Item" maxOccurs="unbounded">
    <xsd:complexType>
      <xsd:attribute name="itemNumber"
type="xsd:integer" use="required"/>
      <xsd:attribute name="itemQty"
type="xsd:integer" use="required"/>
    </xsd:complexType>
  </xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:sequence>
  <xsd:attribute name="accountNumber" type="xsd:integer"
use="required"/>
  <xsd:attribute name="totalAmount" type="xsd:float"
use="required"/>
  <xsd:attribute name="emailAddress" type="xsd:string"
use="required"/>
  <xsd:attribute name="orderType" use="required">
    <xsd:simpleType>
      <xsd:restriction base="xsd:NMTOKEN">
        <xsd:enumeration value="PO"/>
        <xsd:enumeration value="Credit"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:attribute>
  <xsd:attribute name="cardType" use="required">
    <xsd:simpleType>
      <xsd:restriction base="xsd:NMTOKEN">
        <xsd:enumeration value="MC-VISA"/>
        <xsd:enumeration value="Discover"/>
        <xsd:enumeration value="AMEX"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:attribute>
</xsd:sequence>
```

```
        </xsd:restriction>
    </xsd:simpleType>
</xsd:attribute>
</xsd:complexType>
</xsd:element>
</xsd:schema>
```

- 直接使用 SchemaType 定义 OrderStatus 类型。使用 XML schema 枚举出有效的、代表状态的字符串。

```
<TypeDeclaration Id="OrderStatus" Name="OrderStatus">
  <SchemaType>
    <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
      elementFormDefault="qualified" attributeFormDefault="unqualified">
      <xsd:element name="Status">
        <xsd:simpleType>
          <xsd:restriction base="xsd:NMTOKEN">
            <xsd:enumeration value="ValidData"/>
            <xsd:enumeration value="InvalidData"/>
            <xsd:enumeration value="Accept"/>
            <xsd:enumeration value="BadCredit"/>
            <xsd:enumeration value="OverLimit"/>
            <xsd:enumeration value="BadDataFormat"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
    </xsd:schema>
  </SchemaType>
</TypeDeclaration>
```

- 使用 ExternalReference 参考 Order schema 中的 cardType 属性，来定义 CardType 类型：

```
<TypeDeclaration Id="CardType" Name="CardType">
```

```
<ExternalReference location="C:\wfmc\sample\orderschema.xsd"
xref="cardType" namespace="orderschema/Order" />

</TypeDeclaration>
```

- 使用 `ExternalReference` 参考在 WSDL 文档中定义的数据类型，来定义 `CreditInfo` 类型:

```
<TypeDeclaration Id="CreditInfo" Name="CreditInfo">

  <ExternalReference location="C:\wfmc\sample\creditService.wsdl"
xref="CreditInfo" />

</TypeDeclaration>
```

在 WSDL 文档中，`CreditInfo` 类型的定义如下:

```
<types>

  <schema xmlns="http://www.w3.org/2001/XMLSchema">

    <element name="CreditInfo">

      <complexType>

        <sequence>

          <element name="MerchantNumber" />

          <element name="AccountNumber" />

          <element name="Amount" />

          <element name="CardType" />

        </sequence>

      </complexType>

    </element>

  </schema>

</types>
```

### 8.3. ExtendedAttributes

workflow 开发者定义了几个 `ExtendedAttributes` 来扩展 XPD L。为了方便检查，复杂的 `ExtendedAttributes` 在开发者自己的 schema 中定义，此 schema 包含 XPD L schema。

● 用一个 `ExtendedAttribute` 来对活动进行标记，把活动标记为系统活动 (`SystemActivity`)，或者由 workflow 系统实现的活动。在本例中，对 3 个活动进行了标记：`email` 活动、`alarm` 活动、和 `Web service` 活动：

```
<ExtendedAttribute Name="SystemActivity" Value="WebService"/>
<ExtendedAttribute Name="SystemActivity" Value="Email"/>
<ExtendedAttribute Name="SystemActivity" Value="Alarm"/>
```

● 使用一个 `ExtendedAttribute` 来指示活动的位置：

```
<ExtendedAttribute Name="Coordinates">
  <Coordinates xpos="381" ypos="316"/>
</ExtendedAttribute>
```

● 用 `ExtendedAttribute` 为 `Email` 活动提供信息。假定，一部分信息是由建模工具提供的。注意：workflow 开发者可以引用字符串中变量名有前缀“%%”的 workflow 相关数据。

```
<ExtendedAttribute Name="Email">
  <Email to="%%emailAddress" subject="%%orderStatus">
    <Attachments>
      <Attachment>%%docURI</Attachment>
    </Attachments>
    <MessageText>Order number %%orderNumber is %%orderStatus.
    Thank-you for ordering from PQR Products, Inc.</MessageText>
  </Email>
</ExtendedAttribute>
```

## 8.4. External References

本例中使用对一个 WSDL 文档 (`creditService.wsdl`) 的外部参考，定义处理使用信用卡结帐的应用程序。

```
<Application Id="getCreditAuthorization">
  <ExternalReference location="C:\wfmc\sample\creditService.wsdl"
    xref="GetCreditAuthorization"/>
</Application>
```

在 WSDL 文档中此应用程序定义如下:

```
<message name="creditInput">
  <part name="CreditInfo" element="tns:CreditInfo"/>
</message>
<message name="creditOutput">
  <part name="status" type="string"/>
</message>
<portType name="CreditPortType">
  <operation name="GetCreditAuthorization">
    <input message="tns:creditInput"/>
    <output message="tns:creditOutput"/>
  </operation>
  <operation name="ChargeCreditAccount">
    <input message="tns:creditInput"/>
    <output message="tns:creditOutput"/>
  </operation>
</portType>
```

## 8.5 Sample XPDL

```
<?xml version="1.0" encoding="us-ascii"?>
<Package xmlns="http://www.wfmc.org/2002/XPDL1.0"
  xmlns:xpdl="http://www.wfmc.org/2002/XPDL1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xyz="http://www.xyzeorder.com/workflow"
  xsi:schemaLocation="http://www.wfmc.org/2002/XPDL1.0
  http://www.wfmc.org/standards/docs/xpdl.xsd" Id="0" Name="sample
```

```
workflow process">
  <PackageHeader>
    <XPDLVersion>0.09</XPDLVersion>
    <Vendor>XYZ, Inc</Vendor>
    <Created>6/18/2002 5:27:17 PM</Created>
  </PackageHeader>
  <ConformanceClass GraphConformance="NON_BLOCKED" />
  <Script Type="text/javascript" />
  <TypeDeclarations>
    <TypeDeclaration Id="Order" Name="Order">
      <ExternalReference
location="C:\wfmc\sample\orderschema.xsd" />
    </TypeDeclaration>
    <TypeDeclaration Id="CreditInfo" Name="CreditInfo">
      <ExternalReference
location="C:\wfmc\sample\creditService.wsdl" xref="CreditInfo" />
    </TypeDeclaration>
    <TypeDeclaration Id="CardType" Name="CardType">
      <ExternalReference location="C:\wfmc\sample\orderschema.xsd"
xref="cardType" namespace="orderschema/Order" />
    </TypeDeclaration>
    <TypeDeclaration Id="OrderStatus" Name="OrderStatus">
      <SchemaType>
        <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified">
          <xsd:element name="Status">
            <xsd:simpleType>
              <xsd:restriction base="xsd:NMTOKEN">
                <xsd:enumeration value="ValidData" />
                <xsd:enumeration value="InvalidData" />
                <xsd:enumeration value="Accept" />
                <xsd:enumeration value="BadCredit" />
                <xsd:enumeration value="OverLimit" />
              </xsd:restriction>
            </xsd:simpleType>
          </xsd:element>
        </xsd:schema>
      </SchemaType>
    </TypeDeclaration>
  </TypeDeclarations>
</ConformanceClass>
</Script>
</TypeDeclarations>
</ConformanceClass>
</WorkflowProcess>
```

```
<xsd:enumeration value="BadDataFormat" />
</xsd:restriction>
</xsd:simpleType>
</xsd:element>
</xsd:schema>
</SchemaType>
</TypeDeclaration>
</TypeDeclarations>
<Participants>
  <Participant Id="DBConnection">
    <ParticipantType Type="SYSTEM" />
    <Description>Reference to Database Resource</Description>
  </Participant>
</Participants>
<Applications/>
<DataFields/>
<WorkflowProcesses>
  <WorkflowProcess Id="1" Name="EOrder" AccessLevel="PUBLIC">
    <ProcessHeader/>
    <FormalParameters>
      <FormalParameter Id="orderString" Index="1" Mode="IN">
        <DataType>
          <BasicType Type="STRING" />
        </DataType>
      </FormalParameter>
      <FormalParameter Id="returnMessage" Index="2" Mode="OUT">
        <DataType>
          <BasicType Type="STRING" />
        </DataType>
      </FormalParameter>
    </FormalParameters>
    <DataFields>
```



```
<DataField Id="orderNumber" IsArray="FALSE">
  <DataType>
    <BasicType Type="INTEGER"/>
  </DataType>
  <InitialValue>0</InitialValue>
  <Length>0</Length>
</DataField>
<DataField Id="status" IsArray="FALSE">
  <DataType>
    <DeclaredType Id="OrderStatus"/>
  </DataType>
</DataField>
<DataField Id="orderInfo" IsArray="FALSE">
  <DataType>
    <DeclaredType Id="Order"/>
  </DataType>
</DataField>
</DataFields>
<Participants/>
<Applications>
  <Application Id="transformData">
    <FormalParameters>
      <FormalParameter Id="orderStringIn" Index="1"
Mode="IN">
        <DataType>
          <BasicType Type="STRING"/>
        </DataType>
      </FormalParameter>
      <FormalParameter Id="orderInfo" Index="2"
Mode="OUT">
        <DataType>
          <DeclaredType Id="Order"/>
        </DataType>
      </FormalParameter>
    </FormalParameters>
  </Application>
</Applications>
</ProcessDefinition>
```

```

        </FormalParameter>
    </FormalParameters>
</Application>
<Application Id="checkData">
    <FormalParameters>
        <FormalParameter Id="orderInfo" Index="1" Mode="IN">
            <DataType>
                <DeclaredType Id="Order"/>
            </DataType>
        </FormalParameter>
        <FormalParameter Id="statusOut" Index="2"
Mode="OUT">
            <DataType>
                <DeclaredType Id="OrderStatus"/>
            </DataType>
        </FormalParameter>
    </FormalParameters>
</Application>
<Application Id="checkVendor">
    <FormalParameters>
        <FormalParameter Id="accountNumberIn" Index="1"
Mode="IN">
            <DataType>
                <BasicType Type="INTEGER"/>
            </DataType>
        </FormalParameter>
        <FormalParameter Id="amountIn" Index="2" Mode="IN">
            <DataType>
                <BasicType Type="FLOAT"/>
            </DataType>
        </FormalParameter>
        <FormalParameter Id="statusOut" Index="3"
Mode="OUT">

```

```
<DataType>
  <DeclaredType Id="OrderStatus" />
</DataType>
</FormalParameter>
</FormalParameters>
</Application>
<Application Id="enterOrder">
  <FormalParameters>
    <FormalParameter Id="orderInfoIn" Index="1"
Mode=" IN" >
      <DataType>
        <DeclaredType Id="Order" />
      </DataType>
    </FormalParameter>
    <FormalParameter Id="orderNumber" Index="2"
Mode=" OUT" >
      <DataType>
        <BasicType Type="INTEGER" />
      </DataType>
    </FormalParameter>
  </FormalParameters>
</Application>
<Application Id="composeMessage">
  <FormalParameters>
    <FormalParameter Id="statusIn" Index="1" Mode=" IN" >
      <DataType>
        <DeclaredType Id="OrderStatus" />
      </DataType>
    </FormalParameter>
    <FormalParameter Id="orderNumber" Index="2"
Mode=" IN" >
      <DataType>
        <BasicType Type="INTEGER" />

```

```
</DataType>
</FormalParameter>
</FormalParameters>
</Application>
</Applications>
<Activities>
  <Activity Id="1" Name="Check Data">
    <Implementation>
      <Tool Id="checkData" Type="APPLICATION">
        <ActualParameters>
          <ActualParameter>orderInfo</ActualParameter>
          <ActualParameter>status</ActualParameter>
        </ActualParameters>
      </Tool>
    </Implementation>
    <TransitionRestrictions>
      <TransitionRestriction>
        <Split Type="XOR">
          <TransitionRefs>
            <TransitionRef Id="22"/>
            <TransitionRef Id="23"/>
          </TransitionRefs>
        </Split>
      </TransitionRestriction>
    </TransitionRestrictions>
    <ExtendedAttributes>
      <ExtendedAttribute Name="Coordinates">
        <xyz:Coordinates xpos="183" ypos="389"/>
      </ExtendedAttribute>
    </ExtendedAttributes>
  </Activity>
  <Activity Id="5">
```

```
<Route/>

<ExtendedAttributes>
  <ExtendedAttribute Name="Coordinates">
    <xyz:Coordinates xpos="35" ypos="389"/>
  </ExtendedAttribute>
</ExtendedAttributes>
</Activity>

<Activity Id="6">
  <Route/>
  <TransitionRestrictions>
    <TransitionRestriction>
      <Join Type="XOR"/>
    </TransitionRestriction>
  </TransitionRestrictions>
  <ExtendedAttributes>
    <ExtendedAttribute Name="Coordinates">
      <xyz:Coordinates xpos="755" ypos="315"/>
    </ExtendedAttribute>
  </ExtendedAttributes>
</Activity>

<Activity Id="8" Name="Email Confirmation">
  <Implementation>
    <No/>
  </Implementation>
  <ExtendedAttributes>
    <ExtendedAttribute Name="Coordinates">
      <xyz:Coordinates xpos="657" ypos="312"/>
    </ExtendedAttribute>
    <ExtendedAttribute Name="SystemActivity"
Value="Email"/>
    <ExtendedAttribute Name="Email">
      <xyz:Email to="%%orderInfo.emailAddress"
subject="Order %%orderNumber">
```

```
        <xyz:MessageText>Order number %%orderNumber is
being processed. Thank-you for ordering from PQR Products,
Inc</xyz:MessageText>

        </xyz:Email>

    </ExtendedAttribute>

</ExtendedAttributes>

</Activity>

<Activity Id="9">

    <Route/>

    <TransitionRestrictions>

        <TransitionRestriction>

            <Split Type="AND">

                <TransitionRefs>

                    <TransitionRef Id="1"/>

                    <TransitionRef Id="38"/>

                    <TransitionRef Id="2"/>

                </TransitionRefs>

            </Split>

        </TransitionRestriction>

    </TransitionRestrictions>

    <ExtendedAttributes>

        <ExtendedAttribute Name="Coordinates">

            <xyz:Coordinates xpos="572" ypos="389"/>

        </ExtendedAttribute>

    </ExtendedAttributes>

</Activity>

<Activity Id="10" Name="Check Credit Subprocess">

    <Implementation>

        <SubFlow Id="CreditCheck" Execution="SYNCHR">

            <ActualParameters/>

        </SubFlow>

    </Implementation>

    <TransitionRestrictions>
```

```
<TransitionRestriction>
  <Split Type="XOR">
    <TransitionRefs>
      <TransitionRef Id="26"/>
      <TransitionRef Id="31"/>
    </TransitionRefs>
  </Split>
</TransitionRestriction>
</TransitionRestrictions>
<ExtendedAttributes>
  <ExtendedAttribute Name="Coordinates">
    <xyz:Coordinates xpos="381" ypos="535"/>
  </ExtendedAttribute>
</ExtendedAttributes>
</Activity>
<Activity Id="11" Name="Fill Order
Subprocess">
  <Implementation>
    <SubFlow Id="FillOrder" Execution="ASYNCHR">
      <ActualParameters>
        <ActualParameter>orderNumber</ActualParameter>
        <ActualParameter>orderInfo.orderType</ActualParameter>
        <ActualParameter>orderInfo.emailAddress</ActualParameter>
      </ActualParameters>
    </SubFlow>
  </Implementation>
  <ExtendedAttributes>
    <ExtendedAttribute Name="Coordinates">
      <xyz:Coordinates xpos="653" ypos="389"/>
    </ExtendedAttribute>
```

```
        </ExtendedAttributes>
    </Activity>
    <Activity Id="12" Name="Check Order Type">
        <Route/>
        <TransitionRestrictions>
            <TransitionRestriction>
                <Split Type="XOR">
                    <TransitionRefs>
                        <TransitionRef Id="24"/>
                        <TransitionRef Id="25"/>
                    </TransitionRefs>
                </Split>
            </TransitionRestriction>
        </TransitionRestrictions>
        <ExtendedAttributes>
            <ExtendedAttribute Name="Coordinates">
                <xyz:Coordinates xpos="293" ypos="460"/>
            </ExtendedAttribute>
        </ExtendedAttributes>
    </Activity>
    <Activity Id="17" Name="Transform Data">
        <Implementation>
            <Tool Id="transformData" Type="APPLICATION">
                <ActualParameters>
                    <ActualParameter>orderString</ActualParameter>
                    <ActualParameter>orderInfo</ActualParameter>
                </ActualParameters>
            </Tool>
        </Implementation>
        <TransitionRestrictions>
            <TransitionRestriction>
```



```
<Split Type="XOR">
  <TransitionRefs>
    <TransitionRef Id="40"/>
    <TransitionRef Id="21"/>
  </TransitionRefs>
</Split>
</TransitionRestriction>
</TransitionRestrictions>
<ExtendedAttributes>
  <ExtendedAttribute Name="Coordinates">
    <xyz:Coordinates xpos="102" ypos="389"/>
  </ExtendedAttribute>
</ExtendedAttributes>
</Activity>
<Activity Id="32" Name="Enter Order">
  <Implementation>
    <Tool Id="enterOrder" Type="APPLICATION">
      <ActualParameters>
        <ActualParameter>orderInfo</ActualParameter>
        <ActualParameter>orderNumber</ActualParameter>
      </ActualParameters>
    </Tool>
  </Implementation>
  <Performer>DBConnection</Performer>
  <TransitionRestrictions>
    <TransitionRestriction>
      <Join Type="XOR"/>
    </TransitionRestriction>
  </TransitionRestrictions>
  <ExtendedAttributes>
    <ExtendedAttribute Name="Coordinates">
```

```
        <xyz:Coordinates xpos="510" ypos="389"/>
    </ExtendedAttribute>
</ExtendedAttributes>
</Activity>
<Activity Id="33">
    <Route/>
    <TransitionRestrictions>
        <TransitionRestriction>
            <Join Type="AND" />
        </TransitionRestriction>
    </TransitionRestrictions>
    <ExtendedAttributes>
        <ExtendedAttribute Name="Coordinates">
            <xyz:Coordinates xpos="725" ypos="391"/>
        </ExtendedAttribute>
    </ExtendedAttributes>
</Activity>
<Activity Id="39" Name="Compose RejectionMessage">
    <Implementation>
        <Tool Id="composeMessage" Type="APPLICATION">
            <ActualParameters>
                <ActualParameter>orderNumber</ActualParameter>
                <ActualParameter>-1</ActualParameter>
            </ActualParameters>
        </Tool>
    </Implementation>
    <TransitionRestrictions>
        <TransitionRestriction>
            <Join Type="XOR" />
        </TransitionRestriction>
    </TransitionRestrictions>
```

```

        <ExtendedAttributes>
            <ExtendedAttribute Name="Coordinates">
                <xyz:Coordinates xpos="655" ypos="245"/>
            </ExtendedAttribute>
        </ExtendedAttributes>
    </Activity>
    <Activity Id="41" Name="Check Vendor Account">
        <Implementation>
            <Tool Id="checkVendor" Type="APPLICATION">
                <ActualParameters>
                    <ActualParameter>orderInfo.AccountNumber</ActualParameter>
                    <ActualParameter>orderInfo.TotalAmount</ActualParameter>
                    <ActualParameter>status</ActualParameter>
                </ActualParameters>
            </Tool>
        </Implementation>
        <Performer>DBConnection</Performer>
        <TransitionRestrictions>
            <TransitionRestriction>
                <Split Type="XOR">
                    <TransitionRefs>
                        <TransitionRef Id="27"/>
                        <TransitionRef Id="30"/>
                    </TransitionRefs>
                </Split>
            </TransitionRestriction>
        </TransitionRestrictions>
        <ExtendedAttributes>
            <ExtendedAttribute Name="Coordinates">
                <xyz:Coordinates xpos="381" ypos="316"/>
            </ExtendedAttribute>
        </ExtendedAttributes>
    </Activity>

```

```
        </ExtendedAttributes>
    </Activity>
    <Activity Id="56" Name="Compose Acceptance Message">
        <Implementation>
            <Tool Id="composeMessage" Type="APPLICATION">
                <ActualParameters>
                    <ActualParameter>status</ActualParameter>
                    <ActualParameter>orderNumber</ActualParameter>
                </ActualParameters>
            </Tool>
        </Implementation>
        <ExtendedAttributes>
            <ExtendedAttribute Name="Coordinates">
                <xyz:Coordinates xpos="653" ypos="462"/>
            </ExtendedAttribute>
        </ExtendedAttributes>
    </Activity>
    <Activity Id="58" Name="Raise Alarm">
        <Implementation>
            <No/>
        </Implementation>
        <ExtendedAttributes>
            <ExtendedAttribute Name="Coordinates">
                <xyz:Coordinates xpos="100" ypos="250"/>
            </ExtendedAttribute>
            <ExtendedAttribute Name="SystemActivity"
Value="Alarm"/>
        </ExtendedAttributes>
    </Activity>
</Activities>
<Transitions>
    <Transition Id="1" From="9" To="8"/>

```

```
<Transition Id="2" From="9" To="11"/>
<Transition Id="16" From="11" To="33"/>
<Transition Id="17" From="8" To="33">
  <Condition Type="OTHERWISE"/>
</Transition>
<Transition Id="18" From="33" To="6"/>
<Transition Id="20" From="5" To="17"/>
<Transition Id="21" From="17" To="1"/>
<Transition Id="22" From="1" To="12">
  <Condition>status == "Valid Data"</Condition>
</Transition>
<Transition Id="23" From="1" To="39">
  <Condition>status == "Invalid Data"</Condition>
</Transition>
<Transition Id="24" From="12" To="10">
  <Condition>orderType == "Credit"</Condition>
</Transition>
<Transition Id="25" From="12" To="41">
  <Condition>orderType == "PO"</Condition>
</Transition>
<Transition Id="26" From="10" To="32">
  <Condition>status == "Accept"</Condition>
</Transition>
<Transition Id="27" From="41" To="32">
  <Condition>status == "Accept"</Condition>
</Transition>
<Transition Id="28" From="32" To="9"/>
<Transition Id="29" From="39" To="6"/>
<Transition Id="30" From="41" To="39">
  <Condition>status == "OverLimit"</Condition>
</Transition>
<Transition Id="31" From="10" To="39">
```

```
        <Condition>status == "BadCredit"</Condition>
    </Transition>
    <Transition Id="38" From="9" To="56"/>
    <Transition Id="39" From="56" To="33"/>
    <Transition Id="40" From="17" To="58">
        <Condition Type="EXCEPTION"/>
    </Transition>
    <Transition Id="42" From="58" To="39"/>
</Transitions>
</WorkflowProcess>
<WorkflowProcess Id="2" Name="FillOrder" AccessLevel="PRIVATE">
    <ProcessHeader/>
    <FormalParameters>
        <FormalParameter Id="orderNumber" Index="1" Mode="IN">
            <DataType>
                <BasicType Type="INTEGER"/>
            </DataType>
            <Description>Order number assigned to the
order.</Description>
        </FormalParameter>
        <FormalParameter Id="orderType" Index="1" Mode="IN">
            <DataType>
                <ExternalReference
location="C:\wfmc\sample\orderschema.xsd" xref="orderType"
namespace="orderschema/Order"/>
            </DataType>
        </FormalParameter>
        <FormalParameter Id="emailAddress" Index="1" Mode="IN">
            <DataType>
                <BasicType Type="STRING"/>
            </DataType>
        </FormalParameter>
    </FormalParameters>
```

```
<DataFields>
  <DataField Id="docURI" IsArray="FALSE">
    <DataType>
      <BasicType Type="STRING" />
    </DataType>
    <Description>URI of receipt or invoice.</Description>
  </DataField>
  <DataField Id="orderStatus" IsArray="FALSE">
    <DataType>
      <BasicType Type="STRING" />
    </DataType>
  </DataField>
  <DataField Id="creditInfo" IsArray="FALSE">
    <DataType>
      <DeclaredType Id="CreditInfo" />
    </DataType>
  </DataField>
  <DataField Id="creditStatus" IsArray="FALSE">
    <DataType>
      <BasicType Type="STRING" />
    </DataType>
  </DataField>
</DataFields>
<Participants>
  <Participant Id="Shipper">
    <ParticipantType Type="ROLE" />
    <Description>Order shipper</Description>
  </Participant>
</Participants>
<Applications>
  <Application Id="shipOrder">
    <Description>This application presents a screen that
```

presents order information for the order identified by shipOrder. The user may update the order with any changes such as back order information. It returns a string indicating whether the order is complete or on back order.</Description>

```
<FormalParameters>
  <FormalParameter Id="OrderNumberParam" Index="1"
Mode="IN">
  <DataType>
    <BasicType Type="INTEGER"/>
  </DataType>
</FormalParameter>
  <FormalParameter Id="Status" Index="2" Mode="OUT">
  <DataType>
    <BasicType Type="STRING"/>
  </DataType>
  <Description>The String that describes the status
-- either "Complete" or "Backorder"</Description>
</FormalParameter>
</FormalParameters>
</Application>
<Application Id="charge">
  <Description>Charges the credit card and prepares a
receipt for a credit order</Description>
  <ExternalReference
location="C:\wfmc\sample\creditService.wsdl"
xref="ChargeCreditAccount"/>
</Application>
<Application Id="billAccount">
  <Description>Bills the vendor account</Description>
  <FormalParameters>
    <FormalParameter Id="orderNumberParam" Index="1"
Mode="IN">
    <DataType>
      <BasicType Type="INTEGER"/>
    </DataType>
```



```
        </DataType>
    </FormalParameter>
</FormalParameters>
</Application>
<Application Id="createInvoice">
    <Description>Creates an invoice using the order
information and stores it on a server.</Description>
    <FormalParameters>
        <FormalParameter Id="orderNumber" Index="1"
Mode="IN">
            <DataType>
                <BasicType Type="INTEGER"/>
            </DataType>
        </FormalParameter>
        <FormalParameter Id="docURI" Index="2" Mode="OUT">
            <DataType>
                <BasicType Type="STRING"/>
            </DataType>
        </FormalParameter>
    </FormalParameters>
</Application>
<Application Id="createReceipt">
    <Description>Creates a receiptusing the order
information and stores it on a server.</Description>
    <FormalParameters>
        <FormalParameter Id="orderNumber" Index="1"
Mode="IN">
            <DataType>
                <BasicType Type="INTEGER"/>
            </DataType>
        </FormalParameter>
        <FormalParameter Id="docURI" Index="2" Mode="OUT">
            <DataType>
```

```

        <BasicType Type="STRING" />
    </DataType>
</FormalParameter>
</FormalParameters>
</Application>
<Application Id="cancelOrder">
    <FormalParameters>
        <FormalParameter Id="orderNumberIn" Index="1"
Mode="IN">
            <DataType>
                <BasicType Type="INTEGER" />
            </DataType>
        </FormalParameter>
    </FormalParameters>
</Application>
</Applications>
<Activities>
    <Activity Id="21">
        <Route/>
        <ExtendedAttributes>
            <ExtendedAttribute Name="Coordinates">
                <xyz:Coordinates xpos="62" ypos="389" />
            </ExtendedAttribute>
        </ExtendedAttributes>
    </Activity>
    <Activity Id="22" Name="Billing">
        <Implementation>
            <Tool Id="billAccount" Type="APPLICATION">
                <ActualParameters>
                    <ActualParameter>orderNumber</ActualParameter>
                </ActualParameters>
            </Tool>

```

```
</Implementation>

<Performer>DBConnection</Performer>

<ExtendedAttributes>
  <ExtendedAttribute Name="Coordinates">
    <xyz:Coordinates xpos="347" ypos="435"/>
  </ExtendedAttribute>
</ExtendedAttributes>
</Activity>

<Activity Id="23" Name="Charge Credit Card">
  <Implementation>
    <No/>
  </Implementation>
  <ExtendedAttributes>
    <ExtendedAttribute Name="SystemActivity"
Value="WebService"/>
    <ExtendedAttribute Name="Coordinates">
      <xyz:Coordinates xpos="386" ypos="338"/>
    </ExtendedAttribute>
  </ExtendedAttributes>
</Activity>

<Activity Id="30">
  <Route/>
  <ExtendedAttributes>
    <ExtendedAttribute Name="Coordinates">
      <xyz:Coordinates xpos="613" ypos="389"/>
    </ExtendedAttribute>
  </ExtendedAttributes>
</Activity>

<Activity Id="31" Name="email Receipt or Invoice">
  <Implementation>
    <No/>
  </Implementation>
```

```
<TransitionRestrictions>
  <TransitionRestriction>
    <Join Type="XOR"/>
  </TransitionRestriction>
</TransitionRestrictions>

<ExtendedAttributes>
  <ExtendedAttribute Name="Coordinates">
    <xyz:Coordinates xpos="430" ypos="385"/>
  </ExtendedAttribute>
  <ExtendedAttribute Name="SystemActivity"
Value="Email"/>
  <ExtendedAttribute Name="Email">
    <xyz:Email to="%%emailAddress"
subject="%%orderStatus">
      <xyz:Attachments>
        <xyz:Attachment>%%docURI</xyz:Attachment>
      </xyz:Attachments>
      <xyz:MessageText>Order number %%orderNumber is
%%orderStatus. Thank-you for ordering from PQR Products,
Inc.</xyz:MessageText>
    </xyz:Email>
  </ExtendedAttribute>
</ExtendedAttributes>
</Activity>

<Activity Id="36" Name="Ship Order">
  <Description>View order and enter fulfillment
info</Description>
  <Implementation>
    <Tool Id="shipOrder" Type="APPLICATION">
      <ActualParameters>
        <ActualParameter>orderNumber</ActualParameter>
        <ActualParameter>orderStatus</ActualParameter>
      </ActualParameters>
    </Tool>
  </Implementation>
</Activity>
</ProcessDefinition>
```

```
        </ActualParameters>

    </Tool>

</Implementation>

<Performer>DBConnection and Shipper</Performer>

<Deadline Execution="ASYNCHR">

    <DeadlineCondition>3 days</DeadlineCondition>

    <ExceptionName>notifyException</ExceptionName>

</Deadline>

<Deadline Execution="SYNCHR">

    <DeadlineCondition>5 days</DeadlineCondition>

    <ExceptionName>timeoutException</ExceptionName>

</Deadline>

<TransitionRestrictions>

    <TransitionRestriction>

        <Split Type="XOR">

            <TransitionRefs>

                <TransitionRef Id="11"/>

                <TransitionRef Id="10"/>

                <TransitionRef Id="49"/>

                <TransitionRef Id="50"/>

            </TransitionRefs>

        </Split>

    </TransitionRestriction>

</TransitionRestrictions>

<ExtendedAttributes>

    <ExtendedAttribute Name="Coordinates">

        <xyz:Coordinates xpos="133" ypos="386"/>

    </ExtendedAttribute>

</ExtendedAttributes>

</Activity>

<Activity Id="59" Name="Get Credit Info">

    <Implementation>
```

```

        <Tool Id="getCreditInfo" Type="APPLICATION">
            <ActualParameters>

                <ActualParameter>orderNumber</ActualParameter>

                <ActualParameter>creditInfo</ActualParameter>

            </ActualParameters>

        </Tool>
    </Implementation>
    <Performer>DBConnection</Performer>
    <ExtendedAttributes>
        <ExtendedAttribute Name="Coordinates">
            <xyz:Coordinates xpos="311" ypos="338"/>
        </ExtendedAttribute>
    </ExtendedAttributes>
</Activity>
<Activity Id="60" Name="Create Invoice">
    <Implementation>
        <Tool Id="createInvoice" Type="APPLICATION">
            <ActualParameters>

                <ActualParameter>orderNumber</ActualParameter>

                <ActualParameter>docUri</ActualParameter>

            </ActualParameters>

        </Tool>
    </Implementation>
    <Performer>DBConnection</Performer>
    <ExtendedAttributes>
        <ExtendedAttribute Name="Coordinates">
            <xyz:Coordinates xpos="460" ypos="435"/>
        </ExtendedAttribute>
    </ExtendedAttributes>
</Activity>
<Activity Id="61" Name="Create Receipt">

```

```
<Implementation>
  <Tool Id="createReceipt" Type="APPLICATION">
    <ActualParameters>
      <ActualParameter>orderNumber</ActualParameter>
      <ActualParameter>docUri</ActualParameter>
    </ActualParameters>
  </Tool>
</Implementation>
<Performer>DBConnection</Performer>
<ExtendedAttributes>
  <ExtendedAttribute Name="Coordinates">
    <xyz:Coordinates xpos="461" ypos="338"/>
  </ExtendedAttribute>
</ExtendedAttributes>
</Activity>
<Activity Id="63" Name="Raise Alarm">
  <Implementation>
    <No/>
  </Implementation>
  <ExtendedAttributes>
    <ExtendedAttribute Name="Coordinates">
      <xyz:Coordinates xpos="83" ypos="271"/>
    </ExtendedAttribute>
    <ExtendedAttribute Name="SystemActivity"
Value="Alarm"/>
  </ExtendedAttributes>
</Activity>
<Activity Id="64" Name="Cancel Order">
  <Description>View order and enter fulfillment
info</Description>
  <Implementation>
    <Tool Id="cancelOrder" Type="APPLICATION">
```

```
<ActualParameters>

  <ActualParameter>orderNumber</ActualParameter>

</ActualParameters>

</Tool>

</Implementation>

<Performer>DBConnection</Performer>

<ExtendedAttributes>

  <ExtendedAttribute Name="Coordinates">

    <xyz:Coordinates xpos="174" ypos="271"/>

  </ExtendedAttribute>

</ExtendedAttributes>

</Activity>

</Activities>

<Transitions>

  <Transition Id="8" From="21" To="36"/>

  <Transition Id="10" From="36" To="22">

    <Condition>orderType == "PO"</Condition>

  </Transition>

  <Transition Id="11" From="36" To="59">

    <Condition>orderType == "Credit"</Condition>

  </Transition>

  <Transition Id="13" From="60" To="31"/>

  <Transition Id="14" From="61" To="31"/>

  <Transition Id="15" From="31" To="30"/>

  <Transition Id="43" From="59" To="23"/>

  <Transition Id="44" From="23" To="61"/>

  <Transition Id="45" From="22" To="60"/>

  <Transition Id="49" From="36" To="63">

    <Condition>notifyException</Condition>

  </Transition>

  <Transition Id="50" From="36" To="64">
```



```
        <Condition>timeoutException</Condition>
    </Transition>
    <Transition Id="51" From="64" To="30"/>
</Transitions>
</WorkflowProcess>
<WorkflowProcess Id="3" Name="CreditCheck" AccessLevel="PRIVATE">
    <ProcessHeader/>
    <FormalParameters>
        <FormalParameter Id="accountNumber" Index="1" Mode="IN">
            <DataType>
                <BasicType Type="INTEGER"/>
            </DataType>
        </FormalParameter>
        <FormalParameter Id="amount" Index="2" Mode="IN">
            <DataType>
                <BasicType Type="FLOAT"/>
            </DataType>
        </FormalParameter>
        <FormalParameter Id="cardType" Index="4" Mode="IN">
            <DataType>
                <DeclaredType Id="CardType"/>
            </DataType>
        </FormalParameter>
        <FormalParameter Id="status" Index="3" Mode="OUT">
            <DataType>
                <DeclaredType Id="OrderStatus"/>
            </DataType>
        </FormalParameter>
    </FormalParameters>
    <DataFields>
        <DataField Id="creditStatus" IsArray="FALSE">
            <DataType>
```

```
<BasicType Type="STRING" />
</DataType>
<Length>0</Length>
</DataField>
</DataFields>
<Participants/>
<Applications>
  <Application Id="setCreditInfo">
    <Description>Creates and initializes a CreditInfo
object.</Description>
    <FormalParameters>
      <FormalParameter Id="accountNumber" Index="1"
Mode="IN">
        <DataType>
          <BasicType Type="INTEGER" />
        </DataType>
      </FormalParameter>
      <FormalParameter Id="amount" Index="2" Mode="IN">
        <DataType>
          <BasicType Type="FLOAT" />
        </DataType>
      </FormalParameter>
      <FormalParameter Id="cardType" Index="3" Mode="IN">
        <DataType>
          <DeclaredType Id="CardType" />
        </DataType>
      </FormalParameter>
      <FormalParameter Id="creditInfo" Index="4"
Mode="OUT">
        <DataType>
          <DeclaredType Id="CreditInfo" />
        </DataType>
      </FormalParameter>
    </FormalParameters>
  </Application>
</Applications>

```

```
        </FormalParameters>
    </Application>
    <Application Id="getCreditAuthorization">
        <Description>Gets credit authorization from a charge card
web service.</Description>
        <ExternalReference
location="C:\wfmc\sample\creditService.wsdl"
xref="GetCreditAuthorization"/>
    </Application>
    <Application Id="setOrderStatus">
        <Description>Converts status returned by credit check to
OrderStatus.</Description>
        <FormalParameters>
            <FormalParameter Id="creditStatus" Index="1"
Mode="IN">
                <DataType>
                    <BasicType Type="STRING"/>
                </DataType>
            </FormalParameter>
            <FormalParameter Id="orderStatus" Index="2"
Mode="OUT">
                <DataType>
                    <DeclaredType Id="OrderStatus"/>
                </DataType>
            </FormalParameter>
        </FormalParameters>
    </Application>
</Applications>
<Activities>
    <Activity Id="48">
        <Route/>
        <ExtendedAttributes>
            <ExtendedAttribute Name="Coordinates">
                <xyz:Coordinates xpos="61" ypos="395"/>
            </ExtendedAttribute>
        </ExtendedAttributes>
    </Activity>
</Activities>
</ProcessDefinition>
</Workflow>
```

```
        </ExtendedAttribute>
    </ExtendedAttributes>
</Activity>
<Activity Id="49" Name="Set Credit Info">
    <Implementation>
        <Tool Id="setCreditInfo" Type="APPLICATION">
            <ActualParameters>
                <ActualParameter>accountNumber</ActualParameter>
                <ActualParameter>amount</ActualParameter>
                <ActualParameter>cardType</ActualParameter>
                <ActualParameter>creditInfo</ActualParameter>
            </ActualParameters>
        </Tool>
    </Implementation>
    <Performer>DBConnection</Performer>
    <ExtendedAttributes>
        <ExtendedAttribute Name="Coordinates">
            <xyz:Coordinates xpos="151" ypos="394"/>
        </ExtendedAttribute>
    </ExtendedAttributes>
</Activity>
<Activity Id="50" Name="Get Credit Authorization">
    <Implementation>
        <Tool Id="getCreditAuthorization"
Type="APPLICATION">
            <ActualParameters>
                <ActualParameter>creditInfo</ActualParameter>
                <ActualParameter>creditStatus</ActualParameter>
            </ActualParameters>
        </Tool>
    </Implementation>
```

```
        <ExtendedAttributes>
            <ExtendedAttribute Name="SystemActivity"
Value="WebService" />
            <ExtendedAttribute Name="Coordinates">
                <xyz:Coordinates xpos="253" ypos="394" />
            </ExtendedAttribute>
        </ExtendedAttributes>
    </Activity>
<Activity Id="52">
    <Route/>
    <ExtendedAttributes>
        <ExtendedAttribute Name="Coordinates">
            <xyz:Coordinates xpos="444" ypos="397" />
        </ExtendedAttribute>
    </ExtendedAttributes>
</Activity>
<Activity Id="62" Name="Set Order Status">
    <Implementation>
        <Tool Id="setOrderStatus" Type="APPLICATION">
            <ActualParameters>
                <ActualParameter>creditStatus</ActualParameter>
                <ActualParameter>status</ActualParameter>
            </ActualParameters>
        </Tool>
    </Implementation>
    <ExtendedAttributes>
        <ExtendedAttribute Name="Coordinates">
            <xyz:Coordinates xpos="361" ypos="394" />
        </ExtendedAttribute>
    </ExtendedAttributes>
</Activity>
</Activities>
```

```
<Transitions>
  <Transition Id="35" From="50" To="62"/>
  <Transition Id="46" From="48" To="49"/>
  <Transition Id="47" From="49" To="50"/>
  <Transition Id="48" From="62" To="52"/>
</Transitions>
</WorkflowProcess>
</WorkflowProcesses>
</Package>
```

## 9. XPDL Schema

This section presents the full Schema for XPDL.

```
<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema targetNamespace="http://www.wfmc.org/2002/XPDL1.0"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xpdl="http://www.wfmc.org/2002/XPDL1.0"
elementFormDefault="qualified" attributeFormDefault="unqualified">

  <xsd:element name="Activities">

    <xsd:complexType>

      <xsd:sequence>

        <xsd:element ref="xpdl:Activity" minOccurs="0"
maxOccurs="unbounded"/>

      </xsd:sequence>

    </xsd:complexType>

  </xsd:element>

  <xsd:element name="Activity">

    <xsd:complexType>

      <xsd:sequence>

        <xsd:element ref="xpdl:Description" minOccurs="0"/>

        <xsd:element ref="xpdl:Limit" minOccurs="0"/>

        <xsd:choice>

          <xsd:element ref="xpdl:Route"/>

          <xsd:element ref="xpdl:Implementation"/>

          <xsd:element ref="xpdl:BlockActivity"/>

        </xsd:choice>

        <xsd:element ref="xpdl:Performer" minOccurs="0"/>

        <xsd:element ref="xpdl:StartMode" minOccurs="0"/>

        <xsd:element ref="xpdl:FinishMode" minOccurs="0"/>

        <xsd:element ref="xpdl:Priority" minOccurs="0"/>

        <xsd:element ref="xpdl:Deadline" minOccurs="0"
maxOccurs="unbounded"/>

      </xsd:sequence>

    </xsd:complexType>

  </xsd:element>

</xsd:schema>
```

```
<xsd:element ref="xpdl:SimulationInformation" minOccurs="0"/>
<xsd:element ref="xpdl:Icon" minOccurs="0"/>
<xsd:element ref="xpdl:Documentation" minOccurs="0"/>
<xsd:element ref="xpdl:TransitionRestrictions" minOccurs="0"/>
<xsd:element ref="xpdl:ExtendedAttributes" minOccurs="0"/>
</xsd:sequence>
<xsd:attribute name="Id" type="xsd:NMTOKEN" use="required"/>
<xsd:attribute name="Name" type="xsd:string"/>
</xsd:complexType>
</xsd:element>
<xsd:element name="ActivitySet">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="xpdl:Activities" minOccurs="0"/>
      <xsd:element ref="xpdl:Transitions" minOccurs="0"/>
    </xsd:sequence>
    <xsd:attribute name="Id" type="xsd:NMTOKEN" use="required"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="ActivitySets">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="xpdl:ActivitySet" minOccurs="0"
maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="ActualParameter" type="xsd:string"/>
<xsd:element name="ActualParameters">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="xpdl:ActualParameter" minOccurs="0"
maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```



```
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
<xsd:element name="Application">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="xpdl:Description" minOccurs="0"/>
            <xsd:choice>
                <xsd:element ref="xpdl:FormalParameters"/>
                <xsd:element ref="xpdl:ExternalReference" minOccurs="0"/>
            </xsd:choice>
            <xsd:element ref="xpdl:ExtendedAttributes" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attribute name="Id" type="xsd:NMTOKEN" use="required"/>
        <xsd:attribute name="Name" type="xsd:string"/>
    </xsd:complexType>
</xsd:element>
<xsd:element name="Applications">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="xpdl:Application" minOccurs="0"
maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
<xsd:element name="ArrayType">
    <xsd:complexType>
        <xsd:group ref="xpdl:DataTypes"/>
        <xsd:attribute name="LowerIndex" type="xsd:NMTOKEN"
use="required"/>
        <xsd:attribute name="UpperIndex" type="xsd:NMTOKEN"
use="required"/>
    </xsd:complexType>
```

```
</xsd:element>

<xsd:element name="Author" type="xsd:string"/>

<xsd:element name="Automatic">
  <xsd:complexType/>
</xsd:element>

<xsd:element name="BasicType">
  <xsd:complexType>
    <xsd:attribute name="Type" use="required">
      <xsd:simpleType>
        <xsd:restriction base="xsd:NMTOKEN">
          <xsd:enumeration value="STRING"/>
          <xsd:enumeration value="FLOAT"/>
          <xsd:enumeration value="INTEGER"/>
          <xsd:enumeration value="REFERENCE"/>
          <xsd:enumeration value="DATETIME"/>
          <xsd:enumeration value="BOOLEAN"/>
          <xsd:enumeration value="PERFORMER"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:attribute>
  </xsd:complexType>
</xsd:element>

<xsd:element name="BlockActivity">
  <xsd:complexType>
    <xsd:attribute name="BlockId" type="xsd:NMTOKEN"
use="required"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="Codepage" type="xsd:string"/>

<xsd:element name="Condition">
  <xsd:complexType mixed="true">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
```

```
        <xsd:element ref="xpd1:Xpression" />
    </xsd:choice>
    <xsd:attribute name="Type">
        <xsd:simpleType>
            <xsd:restriction base="xsd:NMTOKEN">
                <xsd:enumeration value="CONDITION" />
                <xsd:enumeration value="OTHERWISE" />
                <xsd:enumeration value="EXCEPTION" />
                <xsd:enumeration value="DEFAULTEXCEPTION" />
            </xsd:restriction>
        </xsd:simpleType>
    </xsd:attribute>
</xsd:complexType>
</xsd:element>
<xsd:element name="ConformanceClass">
    <xsd:complexType>
        <xsd:attribute name="GraphConformance">
            <xsd:simpleType>
                <xsd:restriction base="xsd:NMTOKEN">
                    <xsd:enumeration value="FULL_BLOCKED" />
                    <xsd:enumeration value="LOOP_BLOCKED" />
                    <xsd:enumeration value="NON_BLOCKED" />
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:attribute>
    </xsd:complexType>
</xsd:element>
<xsd:element name="Cost" type="xsd:string" />
<xsd:element name="CostUnit" type="xsd:string" />
<xsd:element name="Countrykey" type="xsd:string" />
<xsd:element name="Created" type="xsd:string" />
<xsd:element name="DataField">
```

```
<xsd:complexType>
  <xsd:sequence>
    <xsd:element ref="xpd1:DataType" />
    <xsd:element ref="xpd1:InitialValue" minOccurs="0" />
    <xsd:element ref="xpd1:Length" minOccurs="0" />
    <xsd:element ref="xpd1:Description" minOccurs="0" />
    <xsd:element ref="xpd1:ExtendedAttributes" minOccurs="0" />
  </xsd:sequence>
  <xsd:attribute name="Id" type="xsd:NMTOKEN" use="required" />
  <xsd:attribute name="Name" type="xsd:string" />
  <xsd:attribute name="IsArray" default="FALSE">
    <xsd:simpleType>
      <xsd:restriction base="xsd:NMTOKEN">
        <xsd:enumeration value="TRUE" />
        <xsd:enumeration value="FALSE" />
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:attribute>
</xsd:complexType>
</xsd:element>
<xsd:element name="DataFields">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="xpd1:DataField" minOccurs="0"
maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="DataType">
  <xsd:complexType>
    <xsd:group ref="xpd1:DataTypes" />
  </xsd:complexType>
</xsd:complexType>
```

```
</xsd:element>

<xsd:group name="DataTypes">

  <xsd:choice>

    <xsd:element ref="xpdl:BasicType" />

    <xsd:element ref="xpdl:DeclaredType" />

    <xsd:element ref="xpdl:SchemaType" />

    <xsd:element ref="xpdl:ExternalReference" />

    <xsd:element ref="xpdl:RecordType" />

    <xsd:element ref="xpdl:UnionType" />

    <xsd:element ref="xpdl:EnumerationType" />

    <xsd:element ref="xpdl:ArrayType" />

    <xsd:element ref="xpdl:ListType" />

  </xsd:choice>

</xsd:group>

<xsd:element name="Deadline">

  <xsd:complexType>

    <xsd:sequence>

      <xsd:element name="DeadlineCondition" minOccurs="1"
maxOccurs="1" />

      <xsd:element name="ExceptionName" minOccurs="1"
maxOccurs="1" />

    </xsd:sequence>

    <xsd:attribute name="Execution">

      <xsd:simpleType>

        <xsd:restriction base="xsd:NMTOKEN">

          <xsd:enumeration value="ASYNCHR" />

          <xsd:enumeration value="SYNCHR" />

        </xsd:restriction>

      </xsd:simpleType>

    </xsd:attribute>

  </xsd:complexType>

</xsd:element>

<xsd:element name="DeclaredType">
```

```
<xsd:complexType>
  <xsd:attribute name="Id" type="xsd:IDREF" use="required"/>
</xsd:complexType>
</xsd:element>
<xsd:element name="Description" type="xsd:string"/>
<xsd:element name="Documentation" type="xsd:string"/>
<xsd:element name="Duration" type="xsd:string"/>
<xsd:element name="EnumerationType">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="xpd:EnumerationValue"
maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="EnumerationValue">
  <xsd:complexType>
    <xsd:attribute name="Name" type="xsd:NMTOKEN" use="required"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="ExtendedAttribute">
  <xsd:complexType mixed="true">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
      <xsd:any minOccurs="0" maxOccurs="unbounded"/>
    </xsd:choice>
    <xsd:attribute name="Name" type="xsd:NMTOKEN" use="required"/>
    <xsd:attribute name="Value" type="xsd:string"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="ExtendedAttributes">
  <xsd:complexType>
    <xsd:sequence>
```

```
        <xsd:element ref="xpdl:ExtendedAttribute"
            minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="ExternalPackage">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="xpdl:ExtendedAttributes" minOccurs="0" />
        </xsd:sequence>
        <xsd:attribute name="href" type="xsd:string" />
    </xsd:complexType>
</xsd:element>
<xsd:element name="ExternalPackages">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="xpdl:ExternalPackage" minOccurs="0"
maxOccurs="unbounded" />
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
<xsd:element name="ExternalReference">
    <xsd:complexType>
        <xsd:attribute name="xref" type="xsd:NMTOKEN" use="optional" />
        <xsd:attribute name="location" type="xsd:anyURI" use="required" />
        <xsd:attribute name="namespace" type="xsd:anyURI" use="optional" />
    </xsd:complexType>
</xsd:element>
<xsd:element name="FinishMode">
    <xsd:complexType>
        <xsd:choice>
            <xsd:element ref="xpdl:Automatic" />
```

```
        <xsd:element ref="xpdl:Manual" />
    </xsd:choice>
</xsd:complexType>
</xsd:element>
<xsd:element name="FormalParameter">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="xpdl:DataType" />
            <xsd:element ref="xpdl:Description" minOccurs="0" />
        </xsd:sequence>
        <xsd:attribute name="Id" type="xsd:NMTOKEN" use="required" />
        <xsd:attribute name="Index" type="xsd:NMTOKEN" />
        <xsd:attribute name="Mode" default="IN">
            <xsd:simpleType>
                <xsd:restriction base="xsd:NMTOKEN">
                    <xsd:enumeration value="IN" />
                    <xsd:enumeration value="OUT" />
                    <xsd:enumeration value="INOUT" />
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:attribute>
    </xsd:complexType>
</xsd:element>
<xsd:element name="FormalParameters">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="xpdl:FormalParameter" minOccurs="0"
maxOccurs="unbounded" />
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
<xsd:element name="Icon" type="xsd:string" />
```



```
<xsd:element name="Implementation">
  <xsd:complexType>
    <xsd:choice>
      <xsd:element ref="xpdl:No" />
      <xsd:element ref="xpdl:Tool" maxOccurs="unbounded" />
      <xsd:element ref="xpdl:SubFlow" />
ref="xpdl:    </xsd:choice>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="InitialValue" type="xsd:string" />
  <xsd:element name="Join">
    <xsd:complexType>
      <xsd:attribute name="Type">
        <xsd:simpleType>
          <xsd:restriction base="xsd:NMTOKEN">
            <xsd:enumeration value="AND" />
            <xsd:enumeration value="XOR" />
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:attribute>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="Length" type="xsd:string" />
  <xsd:element name="Limit" type="xsd:string" />
  <xsd:element name="ListType">
    <xsd:complexType>
      <xsd:group ref="xpdl:DataTypes" />
    </xsd:complexType>
  </xsd:element>
ref="xpdl:  <xsd:element name="Manual">
    <xsd:complexType />
  </xsd:element>
```

```
<xsd:element name="Member">
  <xsd:complexType>
    <xsd:group ref="xpdl:DataTypes"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="No">
  <xsd:complexType/>
</xsd:element>

<xsd:element name="Package">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="xpdl:PackageHeader"/>
      <xsd:element ref="xpdl:RedefinableHeader" minOccurs="0"/>
      <xsd:element ref="xpdl:ConformanceClass" minOccurs="0"/>
      <xsd:element ref="xpdl:Script" minOccurs="0"/>
      <xsd:element ref="xpdl:ExternalPackages" minOccurs="0"/>
      <xsd:element ref="xpdl:TypeDeclarations" minOccurs="0"/>
      <xsd:element ref="xpdl:Participants" minOccurs="0"/>
      <xsd:element ref="xpdl:Applications" minOccurs="0"/>
      <xsd:element ref="xpdl:DataFields" minOccurs="0"/>
      <xsd:element ref="xpdl:WorkflowProcesses" minOccurs="0"/>
      <xsd:element ref="xpdl:ExtendedAttributes" minOccurs="0"/>
    </xsd:sequence>
    <xsd:attribute name="Id" type="xsd:NMTOKEN" use="required"/>
    <xsd:attribute name="Name" type="xsd:string"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="PackageHeader">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="xpdl:XPDLVersion"/>
      <xsd:element ref="xpdl:Vendor"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

```
<xsd:element ref="xpdl:Created" />
<xsd:element ref="xpdl:Description" minOccurs="0" />
<xsd:element ref="xpdl:Documentation" minOccurs="0" />
<xsd:element ref="xpdl:PriorityUnit" minOccurs="0" />
<xsd:element ref="xpdl:CostUnit" minOccurs="0" />
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="Participant">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="xpdl:ParticipantType" />
      <xsd:element ref="xpdl:Description" minOccurs="0" />
      <xsd:element ref="xpdl:ExternalReference" minOccurs="0" />
      <xsd:element ref="xpdl:ExtendedAttributes" minOccurs="0" />
    </xsd:sequence>
    <xsd:attribute name="Id" type="xsd:NMTOKEN" use="required" />
    <xsd:attribute name="Name" type="xsd:string" />
  </xsd:complexType>
</xsd:element>
<xsd:element name="ParticipantType">
  <xsd:complexType>
    <xsd:attribute name="Type" use="required">
      <xsd:simpleType>
        <xsd:restriction base="xsd:NMTOKEN">
          <xsd:enumeration value="RESOURCE_SET" />
          <xsd:enumeration value="RESOURCE" />
          <xsd:enumeration value="ROLE" />
          <xsd:enumeration value="ORGANIZATIONAL_UNIT" />
          <xsd:enumeration value="HUMAN" />
          <xsd:enumeration value="SYSTEM" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:attribute>
  </xsd:complexType>
</xsd:element>
```



```
<xsd:enumeration value="h"/>
<xsd:enumeration value="m"/>
<xsd:enumeration value="s"/>
</xsd:restriction>
</xsd:simpleType>
</xsd:attribute>
</xsd:complexType>
</xsd:element>
<xsd:element name="RecordType">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="xpdl:Member" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="RedefinableHeader">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="xpdl:Author" minOccurs="0"/>
      <xsd:element ref="xpdl:Version" minOccurs="0"/>
      <xsd:element ref="xpdl:Codepage" minOccurs="0"/>
      <xsd:element ref="xpdl:Countrykey" minOccurs="0"/>
      <xsd:element ref="xpdl:Responsibles" minOccurs="0"/>
    </xsd:sequence>
    <xsd:attribute name="PublicationStatus">
      <xsd:simpleType>
        <xsd:restriction base="xsd:NMTOKEN">
          <xsd:enumeration value="UNDER_REVISION"/>
          <xsd:enumeration value="RELEASED"/>
          <xsd:enumeration value="UNDER_TEST"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:attribute>
  </xsd:complexType>
</xsd:element>
</xsd:element>
```

```
        </xsd:attribute>
    </xsd:complexType>
</xsd:element>
<xsd:element name="Responsible" type="xsd:string"/>
<xsd:element name="Responsibles">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="xpdl:Responsible" minOccurs="0"
maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
<xsd:element name="Route">
    <xsd:complexType/>
</xsd:element>
<xsd:element name="SchemaType">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:any namespace="##other" minOccurs="0"
maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
<xsd:element name="Script">
    <xsd:complexType>
        <xsd:attribute name="Type" type="xsd:string" use="required"/>
        <xsd:attribute name="Version" type="xsd:string"
use="optional"/>
        <xsd:attribute name="Grammar" type="xsd:anyURI"
use="optional"/>
    </xsd:complexType>
</xsd:element>
<xsd:element name="SimulationInformation">
```

```
<xsd:complexType>
  <xsd:sequence>
    <xsd:element ref="xpdl:Cost" />
    <xsd:element ref="xpdl:TimeEstimation" />
  </xsd:sequence>
  <xsd:attribute name="Instantiation">
    <xsd:simpleType>
      <xsd:restriction base="xsd:NMTOKEN">
        <xsd:enumeration value="ONCE" />
        <xsd:enumeration value="MULTIPLE" />
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:attribute>
</xsd:complexType>
</xsd:element>
<xsd:element name="Split">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="xpdl:TransitionRefs" minOccurs="0" />
    </xsd:sequence>
    <xsd:attribute name="Type">
      <xsd:simpleType>
        <xsd:restriction base="xsd:NMTOKEN">
          <xsd:enumeration value="AND" />
          <xsd:enumeration value="XOR" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:attribute>
  </xsd:complexType>
</xsd:element>
<xsd:element name="StartMode">
  <xsd:complexType>
```

```
<xsd:choice>
  <xsd:element ref="xpdl:Automatic" />
  <xsd:element ref="xpdl:Manual" />
</xsd:choice>
</xsd:complexType>
</xsd:element>
<xsd:element name="SubFlow">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="xpdl:ActualParameters" minOccurs="0" />
    </xsd:sequence>
    <xsd:attribute name="Id" type="xsd:string" use="required" />
    <xsd:attribute name="Execution">
      <xsd:simpleType>
        <xsd:restriction base="xsd:NMTOKEN">
          <xsd:enumeration value="ASYNCHR" />
          <xsd:enumeration value="SYNCHR" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:attribute>
  </xsd:complexType>
</xsd:element>
<xsd:element name="TimeEstimation">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="xpdl:WaitingTime" minOccurs="0" />
      <xsd:element ref="xpdl:WorkingTime" minOccurs="0" />
      <xsd:element ref="xpdl:Duration" minOccurs="0" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="Tool">
```



```
<xsd:complexType>
  <xsd:sequence>
    <xsd:element ref="xpdl:ActualParameters" minOccurs="0"/>
    <xsd:element ref="xpdl:Description" minOccurs="0"/>
    <xsd:element ref="xpdl:ExtendedAttributes" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="Id" type="xsd:NMTOKEN" use="required"/>
  <xsd:attribute name="Type">
    <xsd:simpleType>
      <xsd:restriction base="xsd:NMTOKEN">
        <xsd:enumeration value="APPLICATION"/>
        <xsd:enumeration value="PROCEDURE"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:attribute>
</xsd:complexType>
</xsd:element>
<xsd:element name="Transition">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="xpdl:Condition" minOccurs="0"/>
      <xsd:element ref="xpdl:Description" minOccurs="0"/>
      <xsd:element ref="xpdl:ExtendedAttributes" minOccurs="0"/>
    </xsd:sequence>
    <xsd:attribute name="Id" type="xsd:NMTOKEN" use="required"/>
    <xsd:attribute name="From" type="xsd:NMTOKEN" use="required"/>
    <xsd:attribute name="To" type="xsd:NMTOKEN" use="required"/>
    <xsd:attribute name="Name" type="xsd:string"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="TransitionRef">
  <xsd:complexType>
```

```
        <xsd:attribute name="Id" type="xsd:NMTOKEN" use="required"/>
    </xsd:complexType>
</xsd:element>

<xsd:element name="TransitionRefs">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="xpdl:TransitionRef" minOccurs="0"
maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>

<xsd:element name="TransitionRestriction">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="xpdl:Join" minOccurs="0"/>
            <xsd:element ref="xpdl:Split" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>

<xsd:element name="TransitionRestrictions">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="xpdl:TransitionRestriction"
minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>

<xsd:element name="Transitions">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="xpdl:Transition" minOccurs="0"
maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
```

```
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="TypeDeclaration">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:group ref="xpdl:DataTypes"/>
                <xsd:element ref="xpdl:Description" minOccurs="0"/>
                <xsd:element ref="xpdl:ExtendedAttributes" minOccurs="0"/>
            </xsd:sequence>
            <xsd:attribute name="Id" type="xsd:ID" use="required"/>
            <xsd:attribute name="Name" type="xsd:string"/>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="TypeDeclarations">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element ref="xpdl:TypeDeclaration" minOccurs="0"
maxOccurs="unbounded"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="UnionType">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element ref="xpdl:Member" maxOccurs="unbounded"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="ValidFrom" type="xsd:string"/>
    <xsd:element name="ValidTo" type="xsd:string"/>
    <xsd:element name="Vendor" type="xsd:string"/>
    <xsd:element name="Version" type="xsd:string"/>
```

```
<xsd:element name="WaitingTime" type="xsd:string"/>

<xsd:element name="WorkflowProcess">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="xpdl:ProcessHeader"/>
      <xsd:element ref="xpdl:RedefinableHeader" minOccurs="0"/>
      <xsd:element ref="xpdl:FormalParameters" minOccurs="0"/>
      <xsd:element ref="xpdl>DataFields" minOccurs="0"/>
      <xsd:element ref="xpdl:Participants" minOccurs="0"/>
      <xsd:element ref="xpdl:Applications" minOccurs="0"/>
      <xsd:element ref="xpdl:ActivitySets" minOccurs="0"/>
      <xsd:element ref="xpdl:Activities" minOccurs="0"/>
      <xsd:element ref="xpdl:Transitions" minOccurs="0"/>
      <xsd:element ref="xpdl:ExtendedAttributes" minOccurs="0"/>
    </xsd:sequence>
    <xsd:attribute name="Id" type="xsd:NMTOKEN" use="required"/>
    <xsd:attribute name="Name" type="xsd:string"/>
    <xsd:attribute name="AccessLevel">
      <xsd:simpleType>
        <xsd:restriction base="xsd:NMTOKEN">
          <xsd:enumeration value="PUBLIC"/>
          <xsd:enumeration value="PRIVATE"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:attribute>
  </xsd:complexType>
</xsd:element>

<xsd:element name="WorkflowProcesses">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="xpdl:WorkflowProcess" minOccurs="0"
maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

```
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
<xsd:element name="WorkingTime" type="xsd:string"/>
<xsd:element name="XPDLVersion" type="xsd:string"/>
<xsd:element name="Xpression">
    <xsd:complexType mixed="true">
        <xsd:choice minOccurs="0" maxOccurs="unbounded">
            <xsd:any minOccurs="0" maxOccurs="unbounded"/>
        </xsd:choice>
    </xsd:complexType>
</xsd:element>
</xsd:schema>
```