

工作流引擎与 **BPM** 套件 有怎样的差别？

目录

目录.....	2
90 年代的工作流解决方案.....	3
90 年代末的情况.....	4
完美风暴.....	4
企业 BPM.....	5
结束语.....	7
关于 IBM Business Process Manager.....	7

简介

20 世纪 90 年代，工作流供应商尝试定义工作流及其最佳利用方式，因而在市场上造成了不小的困惑。如今，这种情况在 BPM 行业中再度出现。如果请 10 家不同的供应商来定义 BPM 或 BPM 套件，您很有可能会得到 10 种不同的定义，尽管所有这些供应商都使用相同的基本术语来进行解释。组织需要获得帮助，以便理清头绪，了解 BPM 套件为何与众不同、为何更加重要、为何可用于流程改进。

90 年代的工作流解决方案

让我们首先来介绍一些背景信息。90 年代引入的工作流解决方案主要由一个引擎和一种语言组成。许多解决方案还包含某种类型的图形化建模环境，尽管这种环境可能是较为初级的。只有少数解决方案包含更加健壮、基于 UML 的建模工具，但极有可能是专有建模工具。

尽管当时的主要标准（如 WFML 或 XPD L）已经开始融合，但此类建模工具生成的代码通常采用某种专有工作流语言的形式。它们均提供了基于 XML 的原语，使用流程模式来描述工作流。引擎将解释代码并执行工作流。更改或调试通常在代码内进行，而非在建模工具内进行。

在执行时（运行时），管理令牌的许多元素（表示任务）均内置于引擎之中，例如排队和优先级排列。因此，可以更轻松地利用这些引擎和工具编写程序，控制工作流的行为，因为有关哪些类型的对象可以通过流程（例如任务或代表任务的令牌）的基本概念是内置的。

这些引擎也会关注此类任务执行过程中涉及和使用的资源。因此，类似于组织和/或系统等概念都是内置的，引擎将跟踪“用户 X 的收件箱内目前有多少任务”等。

由于这些工作流系统主要在“任务”分配的领域中为开发人员提供协助，“工作流模式”定义了这些任务的移动，因而有必要在这些领域中最充分地利用这些引擎，同时使其他编程活动最小化。这也就是说，在开发以人工流和任务处理为主要需求的流程自动化方案时，工作流引擎可实现最高的杠杆作用。例如，在图像处理和文档管理系统中，自动化流程的重点并非通过流程的业务数据，因为此类数据最初是非结构化的。只需要在流程中附带一个指向文档或图像的链接，然后让工作流引擎处理其他事情。

在存在大量结构化数据的流程中，需要进行大量的自定义编码来构建数据库表或对象模型。如果存在结构化的业务数据，则还需要大量集成，因为流程应用程序必须触及记录系统。因此，尽管工作

流引擎能提供收益，但是如果所有代码中有一定比例的代码需要在此类情况下编写，则它能够带来的收益较少。此外，工作流引擎通常会对这些情况作出存在重大缺陷的假设，假设其了解各种资源，并维护流程的所有状态。但在与其他系统接合的情况下，如果其他系统自行执行流程中的某些部分，则状态将来回传递。处理这种问题需要编写额外的程序。

此外，如果需求对于工作流的关注较少，而是更为关注将多个系统连接在一起，则往往不使用工作流引擎编写应用程序代码要容易得多。否则，混合模式（和语言）通常会带来技能集方面的障碍……更不用说此后的新员工理解程序时的困难。

90 年代末的情况

- 工作流引擎成功理解了移动系统之间的通信相关模式（Petri-net、pi calculus 和其他数学方法）。已经构造了语言来提供对这些模式的高级访问。
- 非结构化信息可轻松附加到工作流引擎管理的任务。
- 运行时执行反映资源的利用率和消耗情况，允许以更高级的方式理解这些成本。

但也出现了一些问题：

- 这些语言较为分散，而且较新。任何给定解决方案都需要了解其他语言（SQL、C++、C#、Java、Perl、HTML 等）。如果了解一种语言，则存在一定的杠杆作用，但了解这些语言的人员极为短缺。因为这属于一种专长，成本极高。
- 工作流引擎仅处理任意给定流程解决方案的工作流模式和任务部分。任意解决方案的其他部分（用户界面、系统界面、对象模型和这些对象的持久层）均未得到处理。

完美风暴

除了纯工作流世界中发生的情况之外，还有其他一些不同的事件帮助塑造了市场：

- 互联网，从最低的层面上受标准支持的互联网推动着系统界面标准化的需求。随着以经济有效的方式开展电子商务的技术能力的出现，这种集成的人员因素开始成为最大的阻碍。不久之后，对于系统界面的标准就已成为必须，首先出现的此类标准采用了 J2EE 和 .NET 的形式，最近又出现了 Web 服务。

- 业务人员越来越关注流程卓越性，Dell 和 GE 等企业在数字领域中降低流程成本结构的同时提高价值方面取得的成功更给他们提供了动力。由于这样的关注点源于 Michael Dell 和 Jack Welch 等业务领导者，因此以业务为中心的问题本质和解决方案成为一种驱动力。业务人员的绩效按照其流程的绩效进行衡量，因此他们希望在定义、开发、理解和更改这些流程方面掌握占据更有力的控制权。最终结果就是要求业务与 IT 之间进一步实现协调。
- 第一波中低层管理业务人员已经成为技术精通者。在很多情况下，这种趋势在持续推动着企业计算的发展，IT 和业务同时负责定义、开发和更改企业应用程序。

企业 BPM

作为一种技术类别，BPM 套件存在的目的在于使企业能够改进流程。就是这样。为此，需要多方面的要素。BPM 套件必须：

- 使所有人都能尽可能轻松地参与改进流程，以便理解流程并协作改进流程。
- 使得在技术能够提升价值的流程部分中，能够尽可能轻松地利用技术。

为了实现这样的目标，需要运用多种技术（往往通过与这些技术的初始设计完全不同的方法），以便使业务人员能够围绕流程解决方案开展积极的协作。

- **工作流引擎**以及对工作流模式的研究很久以前便证明了多种在异构系统间通信的方法。工作流模式为工作在流程中的遍历方式奠定了基础。
- **BPMN 等高度图形化的建模标记**是便于业务人员掌握的一种表达方法，它不同于单纯代码式的 BPEL。图形文字处理器通过相同的方式为大众提供创建电子文档的能力，最好的方法是选择工作流的图形化视图，以便将流程开发分发给负责设计基于新流程的应用程序的业务和技术人员。
- **双向工程**是 BPM 中必不可少的，在第一个版本部署之后，除了技术程序员之外，它使业务程序员和流程程序员也能继续保持参与流程解决方案的发展演进。在最佳的 BPM 套件中，这也就意味着执行引擎在运行时直接操纵模型。对流程执行行为的更改应立即反映在图形模型之中。就工作流/流程技术而言，这是一项重大的技术进步，仅有少数供应商能够提供这样的能力。

- 已经采用**系统界面的标准化**，以允许更为模块化的方法。不仅在界面定义时是如此，在拖放式界面中更换和修改界面时也是如此。现在，系统界面可以与工作流的图形化流程元素彼此融合，类似地，用户界面现已嵌入流程图或流程模型之中。
- **表示层**，即用户界面，例如表单等，在面向人类的流程中是必不可少的，因此大多数 BPM 套件都提供了一种以图形化的方式开发表单的方法。更为高级的系统为业务用户提供了一种通过图形化方法定义必要的表单数据库的方法，同时允许技术人员更改外观。此外，无论技术人员做出了怎样的更改，业务人员都应该始终能够更改表单。这也是建模级的要求。
- 由于每一个流程应用程序都具有通过该应用程序的结构化业务数据，因而 BPM 套件具有内置的**以图形化方法定义业务对象**（或流程变量）的能力，并使大部分（如果不是全部）为流程应用程序定义的业务对象保持“自由”。
- 既然您将利用以流程为中心的方法来描述应用程序，因而套件应该能够预测**特定于流程的报告**的数据需求。时间间隔（或流程步骤之间的延迟）等方面可轻松在图形化模型中定义，而无需具体了解数据库，也不需要保持或访问它们的 SQL 代码。
- 最后，结合 SOA（面向服务架构）使众多基于 Web 的服务能够执行在 BPM 套件内编排和报告的部分或全部流程，流程的另一个系统方面是事件。高级 BPM 系统将人员和系统输入与输出视为与外部业务事件水平相当。同样，**外部事件的定义和关联**是以图形化方式在建模环境中处理的。

结束语

如您所见，BPM 套件背后的推动因素并非仅仅是工作流，而更多的是图形化表单或系统界面的快速应用程序开发。BPM 是对业务需求的响应。交付考虑周全的 BPM 套件时所用到的技术套件包括此前最佳的技术，工作流也是其中之一。

但利用工作流引擎过去不能、将来也不能为您提供与 BPM 套件相同的收益，因为打包各种技术本身就是一种优势。利用 BPM 套件开发应用程序的方法远远超越使用工作流引擎构建应用程序的方法。原因在于对于问题的理解在不断演进，因此各种解决问题所需的技术的集成也随之发展演进。

在过去，拼写检查是一种独立的应用程序，用于在文档中键入文字之后检查其拼写。拼写检查引擎嵌入于文字处理器中的原因不仅仅在于其功能有着本质的不同，还在于将拼写检查器的使用置于其使用方法和使用时机的上下文之中。打印的文档与此相同，但生产力能够得到提高，因为拼写检查体验已置于其他技术的上下文中，因此每项技术的总体有用性和实用性都得到了提升。

BPM 的重点在于您“如何”构建应用程序，而不仅仅在于您构建的是“什么”。使用 BPM 套件构建的应用程序与使用其他许多技术组合构建的应用程序并无不同，包括“编码和工程”方法。而不同的地方是 BPM 套件能提供更出色、更可伸缩的方法来构建这些解决方案。

优势在于开发人员能够如何访问界面——无论是构建 Oracle 接口的 IT 程序员、构建用户界面的业务程序员还是设计工作流的流程程序员。优势在于这些人员能够如何开展协作、应用程序组件的模块化程度如何、在流程执行过程中实现更改所需的技能集和必备知识如何。

优势在于，无论是否使用经过测试的企业级技术（例如工作流），我们都能够在企业应用程序开发中实现同样的生产力提升，这在过去 30 年采用的个人应用程序开发中是无法实现的。

关于 IBM Business Process Manager

IBM Business Process Manager® 是 IBM 推出的 BPM 套件，用于设计、执行和改进流程。IBM BlueworksLive™ 是惟一一款按需、协作式的流程文档工具，使企业能够映射流程、识别问题、排列改进机会的优先级。IBM 独有的共享模型架构是这些产品的核心所在，与竞争解决方案相比，这种架构能显著减少所需时间和工作量。



© 版权所有 IBM Corporation 2010

IBM Corporation Software Group Route 100
Somers, NY 10589
U.S.A.

保留所有权利

IBM、IBM 徽标、ibm.com 和 WebSphere 是国际商业机器公司在美国和/或其他国家（地区）的商标或注册商标。如果上述及其他 IBM 商标词汇在本文中第一次出现时标记了商标符号（® 或 TM），均代表在本文出版之际，它们是 IBM 在美国或其他国家注册的商标或普通法规定的标。此类商标在其他国家/地区也可能是注册商标或普通法规定的商标。可在网络上获取 IBM 商标的最新列表，请查看 ibm.com/legal/copytrade.shtml 的“Copyright and trademark information”部分。

其他公司、产品或服务名称可能是其他公司的商标或服务标志。

IBM 对于本文信息的准确性不承担任何责任，使用该信息的风险由接收人自行承担。本文的信息随时可能变更或更新，恕不另行通知。IBM 可能随时对本文介绍的产品和/或程序做出改进和/或变更，恕不通知。

本出版物中对 IBM 产品或服务的引用不代表它们可用于所有 IBM 运营的国家。



请回收利用