

Pentaho - 开放源码的商业智能平台

技术白皮书

原文版本	1.1.5
原文链接	
翻译	王娜、覃辉
校对	覃辉
Pentaho 中文讨论组 QQ 群: 12635055 论坛: http://www.bipub.org/bipub/index.asp	

Copyright © 2006 Pentaho Corporation. Redistribution permitted. All trademarks are the property of their respective owners.

For the latest information, please visit our web site at www.pentaho.org

Last Modified on March 28, 2006 (原文)

目录

PENTAHO - 开放源码的商业智能平台	1
技术白皮书.....	1
翻译.....	1
王娜、覃辉.....	1
1 摘要	3
2 问题描述	3
3 传统的解决方法以及他们的缺点	4
4 PENTAHO OPEN BI 套件	5
5 开放源码技术	6
6 体系结构	6
6.1 PENTAHO 服务器.....	7
6.2 服务器资源库	10
6.3 预配置安装	10
6.4 PENTAHO DESIGN STUDIO	10
6.5 内嵌体系结构	12
6.6 PENTAHO SOLUTION	12
7 PENTAHO 公司的角色	14
8 总结	14
9 附录A: 词汇表	15
10 附录B - 开放源码软件	16
10.1 额外的基于WEB的资源.....	17

1 摘要

所有组织都希望在业务过程中通过改善效率、有效性和总性能来提高收入，降低成本，达到改善收益的目的，而商业智能 (BI) 软件供应商声称他们有相应技术来满足这种需求。这些软件供应商销售用于构建**解决方案 (Solution)**的产品或工具，但很少关注客户面临的真正问题。客户为了新需求，而不断去联系新的供应商，买进新的工具，聘请新的顾问。最终，公司的 BI initiative 变成了众多相互独立的**解决方案 (Solution)**，其使用了昂贵的 **monolithic** 商业应用的组合，很难维护定制的代码。

使用这种当前的方法，每为解决一个特定问题，就设计一个工具。真实的商业问题必须被分割成许多单独的任务，如报表，分析，数据挖掘，工作流等等。没有应用程序负责初始化，管理，验证或调整结果，最终需要调用人或程序来弥补这些不足。

这个白皮书描述了 Pentaho 商业智能平台：一个面向解决方案 (Solution) 的BI 平台，其将开放源码组件/公开标准和流程驱动引擎集成在一起。它显示了这个BI 平台如何通过合并并改进 BI 和工作流/流程管理，并以开放源码提供这个平台的方式来解决 BI 问题。

2 问题描述

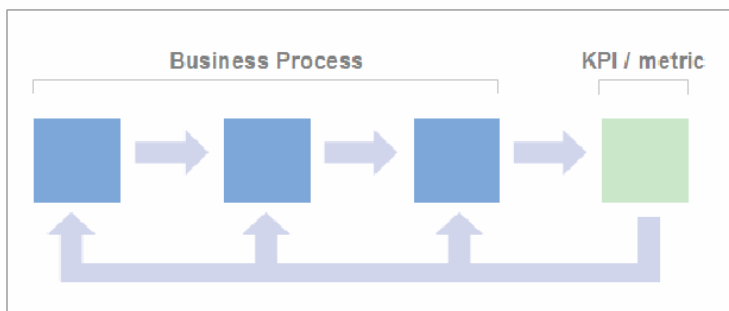
传统的商业智能 (BI) 工具昂贵、复杂，并且在效率和性能方面具有很大不足，难于让企业获得真正益处。各个软件供应商均承诺其 BI 将提供整合，分析和报表等必要功能，将数据转换成蕴涵价值的知识，使管理者得到更及时有用的决策信息。不幸的是，这种 BI 系统和报表系统几乎并没有什么太大的差别，仅仅如此是不能满足需求的。

当传送一个报表，或遇到一个特定情形时，需要触发一些特定的应对操作：重新作出决策，并需要发现引发这些变化的原因，或启动一个特定流程。在这些案例中，信息展示，分析和传送(BI) 是一个较大流程里的一部分。我们需要这样的流程来解决商业问题。(译者注：作者强调业务流程是商业问题的关键。BI只是业务流程的一部分。)

为澄清：

- 一个商业问题的**解决方案 (Solution)** 是一个包含商业智能 (BI) 的流程。
- BI，从它自身来说，很少能成为一个业务问题的完整**解决方案 (Solution)**。
- 如果 BI 是流程的一部分，那么 BI 工具不可避免的也是流程的一部分。
- 一个不理解流程或不知道如何成为流程的组成部分的 BI 工具很难集成进一个更大的**解决方案 (Solution)** 里。

当信息传递到可以使用并利用它影响公司运营的管理人员那里时，一个公司可以最大化其商业价值。当公司优化了很多影响其 **Key Performance Indicator (KPI)**指标 (管理被报表和被分析) 的流程时，公司的关键利益将会得到实现。下面是业务流程与 KPI 的关系图：



BI 实现包含很多阶段：

1. 完成报表和信息传送需求
2. 仪表盘 和 KPI（关键性能指标）显示
3. 异常可能原因的高级分析
4. 集成报表和流程管理/ workflow
5. 那些流程的分析和调整

不幸的是，很多客户仅仅只是实施了报表功能，而没有进行适当的流程管理和 workflow 控制，方法学的缺乏也阻止了他们做更多工作。他们花费了大量资金在私有 **BI 解决方案（Solution）** 上，以期这些软件产品可以解决这些问题。很不幸，传统 BI 工具不能满足这些目标。

3 传统的解决方法以及他们的缺点

一直以来，传统 **BI 解决方案（Solution）** 在以下领域存在很大不足：

价格	软件许可证、维护成本、支持和服务都比较昂贵
可用性	对于大多数用户来说，太难以使用了。
技能	从软件供应商到客户缺少足够的技术转移。缺少执行方法论。
定制性	对于客户来说，太难以开发 解决方案（Solution） 和集成商业规则了。
面向工具集	现有的 solutions 是工具集，而完全不是 解决方案（Solution） 。
扩展性	对于客户和售后供应商来说， 解决方案（Solution） 是私有的，难以扩展和指引系统的运行。客户购买的并不是软件，他们只是为使用它提前付费。这就像租借一辆车，但在第一天就付清了所有租金：这是最糟糕的事情。
报表和分析的焦点	解决方案（Solution） 侧重于 KPIs 的报表和分析上，忽视了影响指标的流程的性能。
流程影响	他们不能确保一个商业流程中不会发生具有驱动作用的变化。他们假设报表的传输将会有影响一个商业流程的副作用。
跟踪与审核	他们不能提供完整的跟踪和审核。谁获得报表？它们将导致什么样的动作？这

	需要花多长时间？是否启动一个商业流程作为结果？这个 流程 要持续多久？ 这个流程的性能如何？
原型	软件定价模型不支持原型阶段，而原型阶段对于确保商业智能项目的成功是非常必要的。重要的财务费用和契约协议必须在系统被完全评估和原型完成之前签订。
许可证弹性和 fit	传统的 BI license models 基于假设：一个指定用户大部分时间以 standalone 应用的方式使用 BI 系统。这个模型不起作用的一个最好实例是 extranet 用户的一个 deployment - infrequent, casual 用户，防火墙之外，此时一个 multi-thousand dollar 的指定用户 license 是无意义的。

4 Pentaho Open BI 套件

Pentaho BI 平台不同于传统的 BI 产品。它是一个以**流程**为中心的，面向**解决方案**（**Solution**）的框架，具有商业智能（BI）组件，使得公司可以开发商业智能问题的完全**解决方案**（**Solution**）。

Pentaho BI 平台，Pentaho Open BI 套件的核心架构和基础，是以**流程**为中心的，因为其中枢控制器是一个工作流引擎。工作流引擎使用流程定义来定义在 BI 平台上执行的商业智能流程。流程可以很容易的被定制，也可以添加新的流程。BI 平台包含组件和报表，用以分析这些流程的性能。

BI 平台是面向**解决方案**（**Solution**）的，因为平台的操作是定义在流程定义和指定每个**活动**的 **action** 文档里。这些流程和操作共同定义了一个商业智能问题的**解决方案**（**Solution**）。这个 BI **解决方案**（**Solution**）可以很容易的集成进平台外部的商业流程。一个**解决方案**（**Solution**）的定义可以包含任意数量的流程和操作。

Pentaho Open BI 套件包含一个 BI 平台，最终用户 BI 能力，和 Pentaho Design Studio:

- BI 平台提供框架和服务的执行，包含日志，审计，安全，调度，ETL，web 服务，属性库和规则引擎。
- 最终用户 BI 能力包含报表，分析，工作流，仪表盘和数据挖掘。
- Pentaho Design Studio是一套设计和管理工具，被集成进广受欢迎的 Eclipse 环境。这些工具允许商业分析人员或开发人员创建报表，仪表盘，分析模型，商业规则和 BI 流程。
- BI 平台和最终用户 BI 能力组成了 Pentaho 服务器。使用Pentaho Design Studio设计 BI **解决方案**（**Solution**），然后发布到 Pentaho 服务器。Pentaho 服务器是运行时引擎，由工作流引擎所驱动，负责协调所有 BI 组件的执行和相互间的通信。

workflow引擎所形成的集中式控制对于 BI 平台的体系结构是很重要的：

- 平台构建于流程和流程定义之上。BI 平台了解流程的本质，因为其中的任何事情均以 一个流程的形式来执行。
- 使用标准的流程定义语言定义流程，这种语言是外部可见的，可编辑的，可定制的， 任何业务逻辑都是公开的，并没有任何隐藏。
- 日志，审计和安全内置于核心中，并被自动使用，以确保总有一个精确的审计跟踪可 用于管理和性能监控。

体系结构是**原创源码**和成熟的开放源码组件的一个结合，他们被集成在一起形成一个完整 的，可伸缩的，复杂 BI 平台。

Pentaho BI 平台构建于服务器，引擎和组件的基础之上。这些提供了系统的 J2EE 服务器， 安全，portal， workflow，规则引擎，图表，协作，内容管理，数据集成，分析和建模功能。这些 组件的大部分是基于标准的，可使用其他产品替换之。

为创建一个**真正集成的、单源的解决方案（Solution）**，Pentaho 增加了以下特征：

- **在解决方案（Solution）定义文档形式上统一的元数据**
- 统一的用户接口和用户接口组件
- 安全性
- Email 和桌面通知
- 所有组件的安装，集成和校验
- 实例**解决方案（Solution）**
- 应用程序连接器
- 使用和诊断工具
- 设计工具
- 个性化定制和配置

5 开放源码技术

Pentaho Open BI 套件使用了很多第三方开放源码组件。体系结构一节显示了 Pentaho 的这 些开放源码组件的体系结构和关系。开放源码组件允许 Pentaho 侧重于系统的增值功能：报表， 分析， workflow集成，商业规则，审计，用户通知和接口，以及 workflow的分析/建模。

有关讨论开放源码软件本质和优势的网站列表请参考附录 B。

作为 Pentaho 在其平台中使用这些开放源码组件的回报，它将 BI 平台对这些项目的贡献 资源和发布组件也纳入进开放源码许可证的控制下。

6 体系结构

为获得如上描述的**解决方案 (Solution)**，Pentaho BI 平台由 Pentaho 服务器，一个基于 Eclipse 的 Design Studio 组成。

6.1 Pentaho 服务器

Pentaho 服务器由一个 BI 平台和传送最终用户 BI 能力的库组成。服务器运行于一个 J2EE 兼容的 web 服务器（如 Apache, JBOSS AS, WebSphere, WebLogic 和 Oracle AS）上。这些也可嵌入进其他服务器或应用程序中。后面的图显示了服务器中较大组件间的关系。

Pentaho 服务器使得 BI 平台的很多功能以一种一致的，熟悉的外观和行为展示给用户。例如，一个组件产生了用户可以访问的报表列表，另一个以日历的方式列出了任务相关的最终期限，第三个显示了用户需要完成的当前任务。每个组件产生的内容和每个用户的角色相关。组件内容可以 XML, HTML 的形式获取或在被包含的 JSR-168 portlets 中显示。Pentaho portlets 可被嵌入进任何支持 JSR 168 标准的 portal, 例如 IBM WebSphere, OracleAS Portal 和 BEA WebLogic Portal。组件用于生成在线报表内容的 XSL 和 CSS 样式单是可访问的，可被完全定制。

Pentaho 服务器包含用于报表，分析，商业规则，email 和桌面通知以及工作流的引擎和组件。这些组件被集成在一起，用于解决商业智能问题。在一个**解决方案 (Solution)**中，每个子系统的行为，相互作用和用户交互被**解决方案 (Solution) 定义文档**的一个集合所定义。

解决方案 (Solution) 定义文档是 XML 文档，它包含：

- 业务流程的定义(XPDL标准)
- 活动的定义，这些活动按需作为部分流程执行，或被 web 服务调用，其包含以下定义：
数据源，查询，报表模板，传送和通知规则，商业规则，**仪表盘和分析视图**。
- 以上所有的项之间的关系

服务器中的组件依赖于一个**解决方案 (Solution)**引擎，可获得可用**解决方案 (Solution)**文档，安全支持，报表，工作流项，数据，和审计信息。

在服务器上可以执行多于一个的**解决方案 (Solution)**。**解决方案 (Solution)**定义文档可从一个服务器复制到另一个，并可被自由分发。

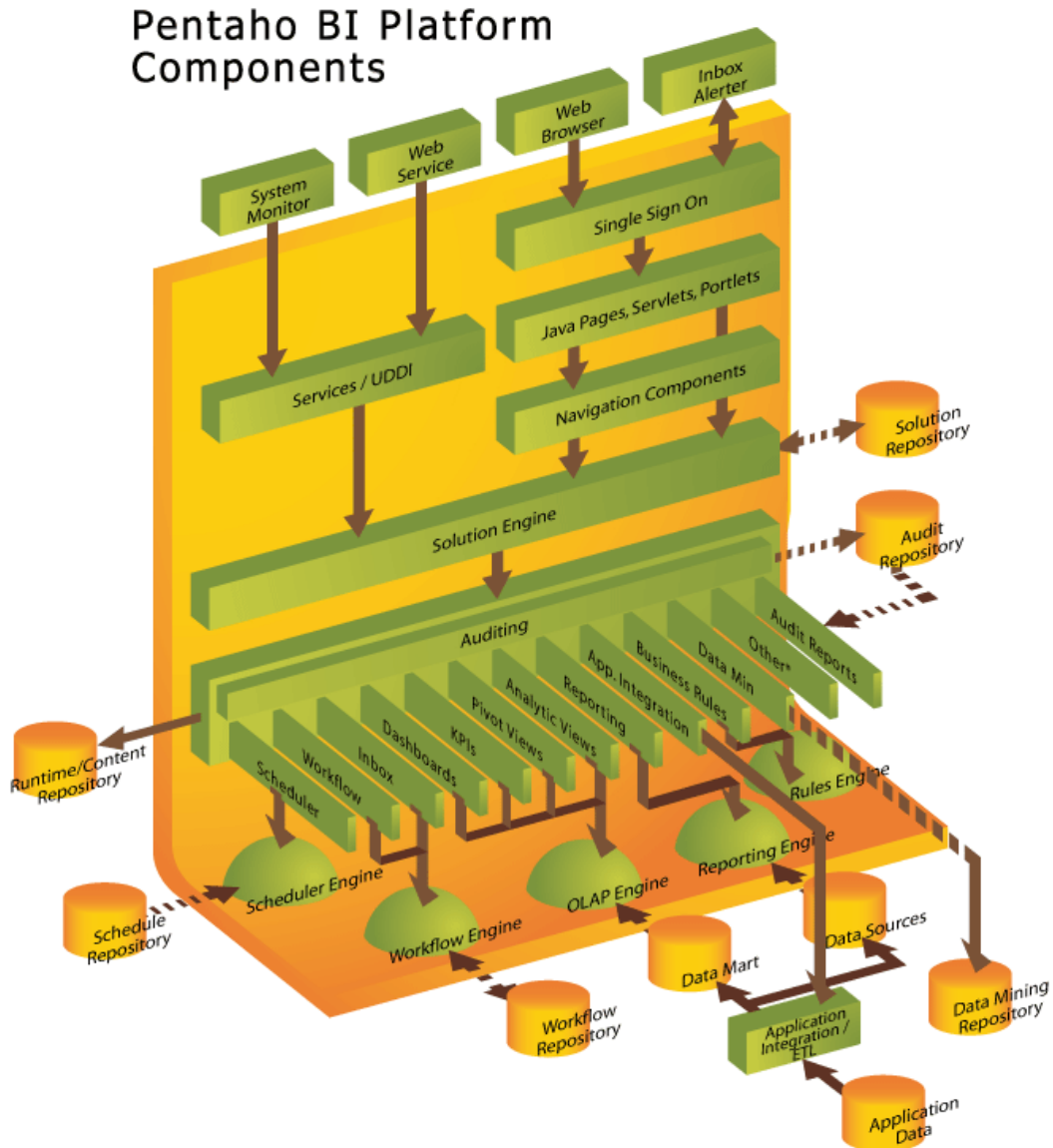
Pentaho 服务器包含了提供高级系统管理的基础设施。这包括系统监控(SMNP)服务，使用报表，Web 服务支持，配置确认工具，和诊断工具。

Pentaho 服务器包含了提供高级流程性能报表和分析的系统和组件。这包括工作流任务上涉及到的工作流项目，单独任务，employees 和 services 上属性的切片和切块 (slice-and-dice)，what-if 和数据挖掘能力。

Pentaho 服务器也支持Enterprise Application Integration (EAI)，用于和operational 应用live

集成，以及 Extract, Transform, and Load (ETL) 能力，用于创建数据仓库和数据集市。

下面是一个服务器体系结构图。



- **解决方案 (Solution)** 引擎对于整个体系结构是很重要的，可以管理对于 BI 组件的访问。
- BI 平台的服务：
 - ◆ 为外部应用程序提供了 web 服务
 - ◆ 可以访问和用户接口组件相同的**解决方案 (Solution)** 引擎
 - ◆ 被工作流引擎和调度器调用以执行系统 **actions**
- 审计内置于平台组件中。通过从工作流和审计库中抽取历史和实时数据，平台提供了

流程性能报表功能。

- 组件是可被添加进系统的模块。
- 每个引擎有相应组件，可以将引擎集成进平台。如果创建了必要的组件的话，引擎可被其他引擎替换，或添加进平台。
- 服务器包括用于构建一个商业智能**解决方案 (Solution)** 所必需的组件和技术：报表， workflow， 商业规则， 仪表盘/分析， web 服务， 调度， 方便的 web 和桌面用户接口的混合， 以及审计。
- Pentaho BI 平台和外部系统集成， 这些系统提供了数据以驱动报表引擎， 并接收来自 workflow 引擎的事件。
- 经由简单网络管理协议(SNMP)， Pentaho BI 平台提供了系统监控能力。
- 资源库被存储进 Pentaho 平台外的一个 RDBMS。预配置安装中内嵌的资源库存储在一个开放源码数据库中， **FireBird** 或 MySQL。如果必须的话， 这些数据库可被其他的关系型数据库代替， 例如 Oracle， SQL服务器或 DB/2。
- Pentaho BI 平台中有多个规则引擎， 因此业务规则被暴露， 并可被简单定制。可在系统中添加额外的规则引擎。对于组件来说， 业务规则引擎是外部的， 且任何组件可以使用任何规则引擎。
- 不是所有组件均显示在图中， 其他（未显示）的组件包括 email， 打印， 消息格式化， workflow 实例属性管理， 流程性能报表和 what-if 分析。
- 体系结构的一些部分使用了技术的组合， 或者使用了易于等价替换的技术：
 - ◆ 被提供的 J2EE 服务器是 JBoss AS， 但是可以使用任何 Java JDK 1.4 兼容的应用程序服务器。
 - ◆ BI 平台提供了内置 Java Server Pages (JSPs), servlets 和 portlets 的用户接口， 可以添加第三方或定制的 JSPs, servlets 或 portlets。
 - ◆ BI 套件包括一个开放源码 OLAP 引擎(**Mondrian**)， 但是可以使用任何 MDX 兼容的 OLAP 服务器(我们计划测试 Microsoft OLAP 服务和 **Hyperion Essbase**)。
 - ◆ 平台提供了一个基于 javascript 的和基于 SQL 的规则引擎， 并基于市场需求， 计划支持 ILOG JRules, Drools 和 Jess。
- Pentaho BI 平台集成和增强了现有的流行的第三方开放源码组件， 例如：
 - ◆ **Mondrian OLAP 服务器**和 **jPivot Analysis Front-End**
 - ◆ **Firebird RDBMS**
 - ◆ **Enhydra ETL, Shark 和 JaWE 工作流**
 - ◆ **Kettle EII and ETL**

- ◆ **JBoss** 应用程序服务器, **Hibernate** 和 **Portal**
- ◆ **Weka** 数据挖掘
- ◆ **Eclipse** 工作台和 **BIRT** 报表组件
- ◆ **JOSSO** 单点登录和 **LDAP** 集成
- ◆ **Mozilla Rhino Javascript Processor**
- BI 平台利用开放标准和协议, 包括:
 - ◆ **XML - W3C** 的可扩展标记语言
 - ◆ **JSR-94 - JCP** 的规则引擎 API
 - ◆ **JSR-168 - JCP** 的 Portlet Spec
 - ◆ **SVG - W3C** 的可扩展矢量图形 (Scalable Vector Graphics)
 - ◆ **XPDL - W3C** 的 XML 流程定义语言
 - ◆ **XForms W3C** 的 Web Forms
 - ◆ **MDX - Microsoft** 的 **OLAP** 查询语言
 - ◆ **WSBPEL - Oasis** 的 Web 服务业务流程执行语言
 - ◆ **WSDL - W3C** 的 Web 服务描述语言
 - ◆ **SOAP - W3C** 的简单对象访问协议

6.2 服务器资源库

Pentaho 服务器包括存储了定义, 执行和审计一个**解决方案 (Solution)** 所必需的数据的内嵌资源库:

- **解决方案 (Solution) 资源库**: 定义了**解决方案 (Solution)** 的元数据。
- **运行时资源库**: workflow 引擎正在管理的工作项目
- **审计资源库**: 跟踪和审计信息

6.3 预配置安装

提供一个预配置的实例部署, 这样平台就可以快速简单的被测试, 配置包括:

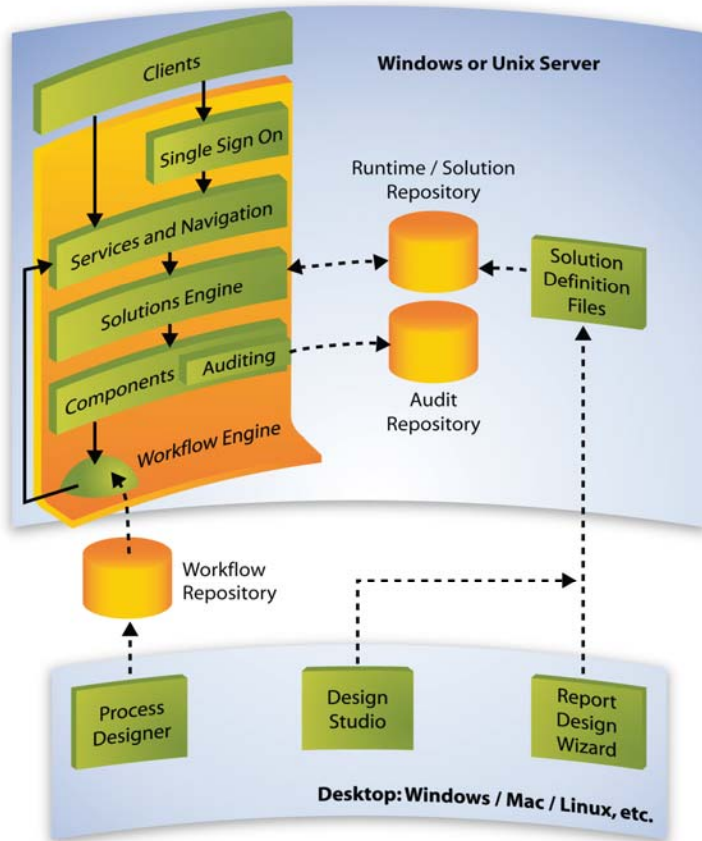
- **JBoss** 应用程序服务器
- **JBoss Portal V2.0**, 一个支持 **JSR-168** 的 portal 服务器
- 实例 **JSPs**, 它演示了平台组件的使用
- 演示例子数据
- 实例报表和 **BI** 流程
- 在实例中使用的用户和角色

6.4 Pentaho Design Studio

设计和管理工作台是一个基于 **Eclipse** 的桌面工作台, 它提供了:

- 报表，仪表盘，分析视图的易于使用的设计工具。
- 工作流设计器
- 商业规则编辑器
- 数据准备的数据挖掘工作台
- OLAP 建模工具

下面提供了一个 Pentaho Design Studio 的体系结构图。



注意 Pentaho Design Studio 体系结构图：

- Design Studio 是一个桌面的，基于 Eclipse 的设计和管理环境。
- Design Studio 产生工作流定义和**解决方案 (Solution)** 定义文件，它们被服务器用于执行 BI **解决方案 (Solution)**。
- Design Studio 运行于多个平台。
- Design Studio 审计**解决方案 (Solution)** 定义文档的创建和编辑。
- 一个版本控制系统用于维护**解决方案 (Solution)** 定义文档，并提供同步和版本管理能力。
- Design Studio 允许以图形化的方式查看和编辑**解决方案 (Solution)**，报表，查询，业

务规则，仪表盘和 workflow。Design Studio 是一个 Java 应用程序，安装在系统管理员和设计者的桌面计算机上。

6.5 内嵌体系结构

- Pentaho 技术可被嵌入进独立的 (standalone) 或基于服务器的 Java 应用程序。
- 必须安装**解决方案 (Solution)** 引擎和组件包。
- 仅有那些组件，引擎和库是配置所必需的。
- 明确来说，这些是可选的：
 - ◆ 工作流引擎，工作流资源库和运行时资源库
 - ◆ 审计和审计资源库
 - ◆ 应用集成/ETL，用于数据抽取，转换和装载
 - ◆ 用户接口组件
 - ◆ **解决方案 (Solution)** 库和**解决方案 (Solution)** 定义文件

6.6 Pentaho Solution

Pentaho BI 平台集成 workflow，商业规则，信息传送和通知，调度，审计，应用程序集成，内容浏览，用户接口，具有报表，分析，仪表盘功能的设计和管理工具，以及数据挖掘组件和引擎。

Pentaho BI 平台的体系结构有很多优点，特定的：

- 通过构建，集成和增强开放源码组件进一个单一的集成平台，BI 实现的成本大幅减少。
- 降低所有权的成本意味着资源可以被调度到别的地方，例如增加实现或定制服务的增值，或扩大商业智能项目的范围，并向终端用户部署更多的高级内容和能力。换句话说，项目预算的很大比例花费在需求收集，实现和服务上，以提高项目成功的可能性。
- 无偿提供软件原型，使得在任何必须的时间段内均可以执行原型。
- 无偿提供软件原型，使得在任何必须的时间段内均可以进行项目需求的迭代。
- 通过构建一个基于 workflow 的平台，提供一个真正的面向服务的体系结构。
- 通过构建一个基于 workflow 的平台，使得易于将 BI 集成进任何商业流程。
- 通过构建一个基于 workflow 的平台，系统就易于集群 (cluster) 和伸缩 (scale)。通过无偿提供软件，不用支付额外的软件成本，系统就可以部署进任意数量的服务器上。
- 通过构建一个基于 workflow 的平台，具有流程性能报表，BI 项目可以不断的调整和改善。
- 通过将信息传送和通知构建进平台，报表，分析，任务和决策要点可以路由到业务流程中涉及到的任何一个人。
- 通过在平台中构建多个规则引擎，所有的业务逻辑都是可定制的。

- 通过将报表，分析和仪表盘构建进平台，激活了 **BI 解决方案 (Solution)** 的混合性的无缝改善。这可以以一种适合于组织的速度来执行，并不被额外的软件成本所限制，或很大程度上由购买周期或 **vendor quarter-closing** 和 **fiscal commitments** 驱动。
- 通过将数据挖掘构建进平台，可以适时的添加最高级的数据分析。
- 通过提供一个 **out-of-the-box** 服务器和一套组件和引擎，就易于部署和集成任何必须的配置。
- 通过将审计和审计报告，系统监控和管理特征集成进平台，系统易于维护。
- 通过提供 **intuitive**，且易于定制的用户接口，使用系统以及培训的花费就会大大降低。

为产生 **Pentaho** 体系结构，如下操作是必要的：

- 定义整个体系结构的需求
- 确定是否设计和构建每个组件，或是使用现有的第三方组件。
- 确定每个组件/项目的提供者
- 在每个组件/项目进行适当的研究
- 安装和配置每个组件
- 为每个组件设计和实现一个集成层
- 设计和实现一致的用户接口组件
- 设计和实现一致的管理工具
- 设计和实现分析和建模工具
- 设计和实现公共服务和基础设施
- 设计和创建库
- 设计和实现新组件或用新功能增强现有组件
- 集成安全
- 集成审计
- 设计和实现流程性能报表
- 创建一种统一定义语言
- 有这类产品开发的相关经历

由于它的体系结构和许可证，使用这个 **BI** 平台作为一个商业智能**解决方案 (Solution)** 的一部分，将：

- 提高用于需求收集的预算和时间
- 提高用于原型的预算和时间
- 提高用于实现和项目首次展示的预算和时间
- 提高用于服务和培训的预算和时间

- 促进实现阶段的开始
- 减少实现**解决方案 (Solution)** 所必需的技能范围

这些因素结合起来，降低了项目的成本和风险，提高了项目成功的可能性。

7 Pentaho 公司的角色

Pentaho 公司促进和管理 Pentaho BI 平台的开发。Pentaho 提供了源码，提供全职工程和服务人员，并管理项目的 roadmap。具体说来，Pentaho:

- 为开放源码团体构建组件
- 为内嵌的开放源码项目作贡献
- 增强其他人开发的组件
- 将组件集成进内聚的，灵活的构建块，Java 开发者可以使用构建块快速集成定制**解决方案 (Solution)**。
- 使用这些构建块为终端用户创建完整的，out-of-the-box 产品和一个全面的 BI 平台。
- 提供全面的培训，技术支持，版本管理，质量保证和企业服务
- 经由它的网页和论坛 Fosters 开放源码 BI 协作
- 使用专业的工程学方法
- 使得整个**解决方案 (Solution)** 可用 – 没有隐藏的升级
- 陈述 BI 缺点和开放源码的关注点

8 总结

Pentaho BI 平台解决了现有的私有 BI 产品所具有的不足:

价格	通过使用开放源码，Pentaho 能显著降低 BI 解决方案 (Solution) 所有者的成本。开放源码组件协助提供一个可扩展的，健壮的平台。所有者的成本主要和适当的集中于 解决方案 (Solution) 的定制和业务流程的集成/自动化。
可用性	BI 平台有对任务敏感的 (intuitive) 的收件箱，它和外部系统紧密集成，提供用户接口的彻底定制。很容易使用 解决方案 (Solution) 的结果。
技能	BI 平台使用经过验证的标准，因此可以使用第三方工具和 portable skills。
定制性	BI 平台有外部规则引擎，可编辑的流程，源代码可用性，可重用的 BI 组件，以及充分定制的用户接口。平台执行可编辑的流程，并可和外部业务流程集成。它也在任何可能的地方 leverages 定制和集成的开放标准。结果是一个高度可定制的 解决方案 (Solution) 。
面向工具集	BI 平台是以流程为中心的，并面向 解决方案 (Solution) 。它不是被构建成为一个工具集，而是一个完全内嵌的，易于集成的系统。

扩展性	提供了源代码。基础设施, 组件, 引擎和接口可任意修改。在许可证购买上节省下来的资金可用来扩展任何必需领域上的平台。平台特征的 roadmap 是公开的, 并可发起特征开发。
报表和分析的焦点	BI 平台是以流程为中心的, 基于工作流的, 以商业规则为指导。它可以很容易的集成进业务流程, 并有内置的流程性能报表。业务规则用于识别问题和异常, 并执行被审计的流程来解决它们。
流程影响	BI 平台跟踪信息传送, 并允许收件人启动定制的业务流程。报表的传送不是‘终点 (deadend)’。产生流程绩效报告, 进行流程建模。可以依据基本指标, 采取措施来测量流程的有效性。
跟踪与审核	BI 平台包括健壮的以流程为中心的跟踪和审核。可以访问一个状况的历史, 包括要采取的措施 (action), 讨论/注释, 和支持信息。
原型	BI 平台许可证允许只要不会使用很大花费, 就尽量构造原型。
许可证弹性和 fit	BI 平台许可证是弹性的, 反映了真实世界的客户价值。BI 是较大的商业流程的一部分, 有增值。除了这个, 在一个合理的成本上的基于 CPU 的许可证使得 extranet 应用 (对于传统的基于 role 的, named-user-only licensing 是不现实的) 有了可能性。

9 附录 A: 词汇表

- BI: 商业智能。IT 市场的一部分, 包括报表和分析应用程序。
- BPEL: 商业流程执行语言。一个标准的系统, 用于协调多个服务上的工作流。web 服务被称作 BPEL4WS 或 WSBPEL。
- CMS: Content Management System, 内容管理系统。
- HTML: Hyper Text Markup Language, 这个语言用于创建网页。
- IBM WebSphere™: IBM 公司的私有 portal, 支持 JSR 168 标准。
- J2EE: Java 2 企业版。一个平台独立的标准, 用于有标准组件的多层企业软件。
- Java™: 编写软件的平台独立的语言。
- JBoss™ : 一个专业的开放源码公司, 提供了很多开放源码技术, 以及支持和服务。
- JBoss™ Portal: JBoss 公司的开放源码 portal, 支持 JSR 168 标准。
- Jetspeed: Apache 基金会的开放源码 portal, 支持 JSR 168 标准。
- JSR 94: Java Rule Engine API。一个标准规范, 用于基于 java 的规则引擎。
- JSR 168: Portlet 规范。一个标准规范, 用于为内嵌进一个 portal, 创建用户接口。
- JSR 170: 用于 Java 技术 API 的内容库, 用于实现内容管理系统的一个标准。

- Kerberos: 一个标准的网络认证协议。
- KPI: Key Performance Indicator。关键性能指标。商业所追踪的一个性能指标。
- LDAP: Lightweight Directory Access Protocol。轻量级目录访问协议。一个标准协议，用于访问资源属性，例如 employees 或 web 服务。
- Metadata: 用于描述数据的结构和内容的信息。
- MS Excel™ Add-In: 在 MS Excel™ 中执行的一个 Windows™ 程序。
- Notification: Tray。一般称作 System Tray。显示警告和服务的 Windows™任务栏区域。
- OLAP: 在线分析处理。一个应用程序，允许浏览，报表和分析多维数据。
- OracleAS Portal™: Oracle 公司的私有 portal，支持 JSR 168 标准。
- Portal: 一个基于 web 的应用程序，用于集成来自不同系统的内容，在一个或多个多功能页面中显示。
- Portlet: 一个 portal 的一个组件，它为来自另一系统的 portal 页面提供了内容。
- Third Party Portal: 一个 portal 服务器，它不是内嵌进 Pentaho BI 平台的 portal 服务器。
- WSRP: Web Services for Remote Portlets。一个规范，允许一个 portal 调用另一个 portal，获得要显示的内容。
- XML: eXtensible Markup Language。可扩展标记语言，一个用于创建机器可读数据的标准语言。一般用于传输和转换数据。
- XPDL: XML 流程定义语言。一个标准语言，用于以 XML 形式描述一个流程。
- XSL: Style sheets。一个标准，用于将一个 XML 文档转换成另一种格式，例如 HTML。
- Weblogic Portal™: BEA 公司的私有 portal，支持 JSR 168 标准。

10 附录 B – 开放源码软件

开放源码软件运动开始于 1970 年代，正在获得如下所示的重要推动力：

- 公司，例如 IBM 和 Sun 正将先前的私有产品发布进开放源码社区，为增强性能和稳定性的改善。一个主要的例子是 Eclipse，一个开放源码开发工具，是 IBM 捐献给开放源码社区的(原来的 IBM 工作台)。以下是 Eclipse 项目的一些特征，它说明了支持很多成熟开放源码项目的资源种类：
 - ◆ Eclipse 是由许多公司联合支配的，这些公司包括 IBM, Borland, Rational, Red Hat, Sybase, Fujitsu, Hitachi, Oracle, SAP, OMG 和 Intel。
 - ◆ Eclipse 是在北美最受欢迎的 Java 开发环境(45% 市场占有率)，每年的市场增长率是60%，有望成为世界第一位的 Java 工具。

- ◆ Eclipse 在世界范围内有20多个大学为之做贡献，且eclipse.org 有足够的基金来做研究。
- ◆ Eclipse 2005 conference 的 keynote speakers 是 Tim O'Reilly, 来自O'Reilly & Associates publishing (被认为是世界最好的计算机书籍出版商)和Urs Hoelzle (Google 的 first V.P. of Engineering)
- ◆ 60 多个国家的 1,200 多个开发人员正在这个项目上工作。
- world's web servers 的一个最近的(Nov 2004) 调查发现：70% (13.1 million)的用户在使用Apache, 一个开放源码产品，它是唯一一个获得了市场份额的 web 服务器。
- 开放源码产品和包含了或内置了开放源码的产品有：**FlexWiki** (Microsoft), Windows Installer (Microsoft), Windows 模板库(Microsoft), Cloudscape 数据库(IBM - worth \$85 million), StarOffice (Sun), CVL (Intel), Oracle Internet 应用程序服务器, Oracle 企业管理器, **Ingres (Computer Associates)**, SADB (SAP), OS X (Apple)。
- 为和开放源码操作系统竞争, Microsoft 同意向中国政府提供 Windows的源代码。
- Information Week 报告显示被调查公司的 65% 使用开放源码产品, 仅有17% 短期内不想使用开放源码。
- IBM 和 Sun (在多数市场的竞争者)正一起推动开放源码 Java 实现。
- California Governor Arnold Schwarzenegger 承诺进行 California 的 procedures 和 finances 的彻底审计。这个审计导致 **California Performance Review**。在 review 中考虑使用信息技术, California Performance Review 建议 State of California 探索可用的开放源码。(http://www.report.cpr.ca.gov/cprprt/issrec/stops/it/so10.htm)

10.1 额外的基于 Web 的资源

开放源码组织

<http://www.opensource.org/>

O'Reilly 出版社

<http://opensource.oreilly.com/>

开放源码优点和局限性 – 概览

http://www.masternewmedia.org/2002/06/30/open_source_advantages_and_limitations_overview.htm

对于IT 顾问, 开放源码的优点

<http://www.cyber.com.au/misc/frsbiz/consults.htm>

开放源码模型的已知缺点

http://eu.conecta.it/paper/Perceived_disadvantages_ope.html

开放源码软件的优点

http://www.cs.virginia.edu/~pev5b/writing/econ_oss/advantages.html

开放源码 Funding: \$40 million from IBM alone

<http://news.com.com/2100-1001-275388.html?legacy=cnet>

开放源码的十个传奇故事

http://opensource.oreilly.com/news/myths_1199.html

Information Week: Open-Source Software Use Joins The Mix

<http://informationweek.smallbizpipeline.com/trends/51201995>

CIO 杂志: 你的开放源码计划

<http://www.cio.com/archive/031503/opensource.html>

CIO杂志: 这些人是谁?

http://www.cio.com/archive/031503/opensource_sidebar_3.html

开放源码成为主流

<http://www.minoru-development.com/en/opensource.html>