以用户为中心的、表单驱动的工作流

您的应用程序是否需要以一系列预定义的步骤向用户发送电子表单以完成一个工作流过程?

例如，在员工自助帮助台(help-desk)应用程序中，员工填写表单，提交后由电子邮件送到帮助台专家处。问题解决后，完成的表单送回原发送者手中以签字结束。在签字结束后，这个表单就记录在帮助台的跟踪数据库中。

在 Domino 中构建和部署这种类型的解决方案相对容易，因为有内置的电子邮件 API 和灵活的基于角色的标识和访问控制。

一个增值解决方案是将 Lotus Domino 与 WebSphere Application Server 和 Web Sphere MQ 集成，事务进程工作流最适合由 WebSphere 应用程序管理，而为它提供支持的以用户为中心的、基于表单的工作流则使用 Lotus Domino。这种结合了系统事务与人工处理的、端到端的工作流解决方案所具有的能力是很少有单个厂商可以提供的。

对于内容管理审批工作流，Lotus Domino 是一个非常好的选择，因为它有灵活的工作流能力以及 WebSphere Portal Content Publisher。WebSphere Portal Content Publisher 有内置的批准工作流例程、并且可以利用 Lotus Workflow 或者 IBM Content Manager 工作流。

访问控制

用户是否需要委派他们的行动?

一些应用程序要求有一种授权功能需求，可以使一位用户授予另一位用户访问控制权限。常用于让一位用户代表另一位用户采取行动，并跟踪是由哪一位用户完成的行动。一个典型的例子是经理将责任委派给一位助手。这位助手不会得到成为经理的所需要的访问权限，但是应该得到代表经理采取行动所需要的访问权限。不同之处是由助手做进行的所有创建/读取/更新/删除行动与由经理所进行的这些行动是有区别地记录的以便进行审查。这种功能是与使用基于角色标识的底层数据存储和访问数据库中单独元素的支持紧密相关的。 Lotus Domino 支持基于角色的标识和比 RDB 更细化的访问控制模型，使得在 Domino 中实现这种类型的应用程序更容易。

用户的访问控制是否需要限制为表中记录或者表单中字段的一个子集?

例如，经理可以访问“员工工资”表，但是他只能看到他所管理的员工的记录。同样，一个填写表单的用户可能只能看到部分字段，而使用同一个表单的经理可以在其视图中看到更多字段。

Lotus Domino 有一个独特的安全模型，在这个模型中，访问控制可以施加到应用程序和数据的非常小的子集上。还可以在运行中以编程方式改变访问控制，根据一系列动态条件对各人所能看到的内容进行调整。

应用程序是否需要动态的、基于角色的用户标识?

大多数应用程序在公司用户目录中使用组。不过，在许多情况下组是应用程序特定的。将应用程序特定的组储存在公司目录中是不明智的，因为牵涉到管理，所以更好的解决方案是将用户角色结合到应用程序中，将角色储存为应用程序数据的一部分并编写确定当前用户角色的逻辑。

审批过程是需要角色的应用程序的一个好例子。例如，在人力资源(HR)招聘应用程序中，需要审批工资标准的人是人力资源经理、财务经理和雇用经理。他们是目录中不同组的成员，但是工资批准权限不能授予这些组，因为它们包含了很多的其他人。不能创建一个名为 Salary Approval 的新组，因为在每次招聘时雇用经理成员都不一样。因此解决方案就是创建一个应用程序特定的角色，用户可以加入和退出这个角色。角色成员关

系是在运行时通过应用程序逻辑改变的，而管理是在应用程序级别而不是在(系统的)目录级别进行的以降低应用程序管理开销。

虽然可以在许多应用程序平台上构建一个基于角色的用户标识模型，但是在 Domino 应用程序中这是一个简单和常见的做法。这种能力还为许多需要动态角色关联的 J2EE 应用程序提供了 Lotus Domino 与 WebSphere Application Server 或者 WebSphere Portal 的另一个集成点。

应用程序生存周期

战略与战术

应用程序的战略性如何? 或者它是一个战术的或者局部解决方案?

这个问题需要您对应用程序的目的和生存周期进行判断。如果预计应用程序有5年或者更长的生存周期，或者如果应用程序对于支持核心业务是至关重要的，那么它就是战略性的。战术或者局部解决方案支持相对短期的项目或者启动业务(initiative)。这类解决方案可能还包括不常使用的应用程序。例如，一个一年只使用一次的 HR 员工表现评审应用程序是一个战术或者局部解决方案，因为它不常使用并与核心业务操作不直接相关。

作为平台及其组件不断增长的集成的结果，出现了快速应用程序开发(RAD)环境。大多数 RAD 环境使得所构建的应用程序的全部灵活性和未来扩展具有高度易用性。由于 Lotus Domino Designer 的 RAD 本性，大多数 Domino 应用程序是短期或者战术的解决方案，因为开发人员可以在相对短的时间内构建和部署应用程序。如果您的组织有这种部署在 Lotus Domino 上的应用程序，那么不建议在 J2EE 上重新构建它们，因为很可能您得到的好处不足以抵消开发成本。

战略性应用程序设计为在应用程序生存周期内给予最大回报，这包括无法预见的应用程序未来扩展和集成。仔细评估应用程序的生存期并考虑在本文中描述的因素可以帮助确定战略性解决方案的合适平台。

实现价值的时间

您的应用程序是否需要在 6 个月内完成设计或者增强、开发、测试和部署?

例如，根据内部分析，一个部门有三个月的时间来实现变化管理系统以达到 ISO 9000 的要求。这个开发周期并不长，但是不解决它，部门就不能达到标准。如前所述，Lotus Domino 的一个主要优点是快速的应用程序开发环境。可以在数周内而开发、测试和部署一个应用程序，而不像其他应用程序服务器那样需要数月。当然，"里程数"可能有所不同，这取决于应用程序的复杂程度。许多非常适合 Lotus Domino 的应用程序构建都相对较简单，特别是当开发人员使用 Lotus Domino 自带的标准系统模板之一时。

环境因素

IT 企业基础设施标准

应用程序是否服从 IT 战略政策或者企业控制?

许多 IT 组织对他们的应用程序基础设施做出战略决策，从而简化系统管理并通过服务器整合节省资金。大多数公司将他们的战略建立在 J2EE 应用程序体系结构、使用 LDAP 的目录服务、使用 RDB 的数据存储的基础之上。如果您的应用程序体系结构是由这种 IT 企业战略控制的，那么它很可能属于 WebSphere 平台，记住可能需要 Lotus Domino 提供 J2EE 体系结构中没有的功能。另一个要记住的事实是可以将 Domino Directory 配置为一个 LDAP 目录，这样您现有的 Domino Directory 就可以加入到 IT 政策、IT 基础设施和支持资源中。

是否可以使用全部 IT 资源?

应用程序开发、部署和管理总是需要一定的开销，包括 IT 基础设施(数据库、应用程序服务器、目录和安全)和

用于系统管理、数据库管理、目录服务和安全体系结构的人力资源。

在许多情况中，只有两种情况有理由需要在缺少可用 IT 基础设施和人力资源的条件下使用应用程序处理业务。考虑一个需要部署到遥远的、不发达地区的应用程序，那里没有足够的本地 IT 支持或者广域网(WAN)带宽。除了这种基础设施限制，预算限制是另一个要考虑的因素。例如，一个业务单位本来需要有一个解决方案，但是不能得到 IT 资源所需要的足够资金。

Lotus Domino 是这两种情况的最合适的解决方案，因为它是完全集成的、独立的应用程序服务器。相反，WebSphere 应用程序平台需要在环境中有各种其他基础设施元素，如需要在应用程序服务器之外另外安装和管理一个 LDAP 目录和关系数据库，则需要更多 IT 资源以支持这个平台。

应用程序开发技能

您是否可以获得或者可以使用 Lotus Notes 或者 Java 开发和管理技能?

许多应用程序需要有开发人员去创建和维护，并要有系统管理员管理它们。应用程序体系结构常常是由组织可以使用的技能来决定的。判断哪些资源是当前可用的、哪些是可以通过雇用或者外包而获得的是决定选择哪种平台的重要因素。尽管现场可用的技能类型可以作为不错的出发点，但保持技能不落伍也很重要，这还包括对技能的交叉培训。Domino 开发人员会因对 WebSphere 的价值和可以起的作用的理解而获益，同样，WebSphere 开发人员会因为对 Domino 带给应用程序的协调能力的理解而获益。

伸缩性要求

用户数量

并发 Web 浏览器用户数量是否超过 5000?

有几种因素决定 Web 应用程序的伸缩性要求，其中之一是应用程序的并发活跃用户数量。作为一个一般性规则，如果并发活跃用户数超过 5000 并且他们使用 Web 浏览器，那么应用程序最好使用 WebSphere Application Server。

不过，如果应用程序是为 Lotus Notes 客户提供服务的，那么就选择 Lotus Domino。Lotus Notes 使用客户/服务器体系结构，在这里很多进程是在客户端完成的，以使服务器具有更大的伸缩性。

伸缩性的基础设施配置

您的应用程序需要横向伸缩还是纵向伸缩，还是两者都需要?

使用适当的硬件配置是满足应用程序伸缩性和可用性需求的关键。水平伸缩指的是让应用程序分布到几个物理服务器或者几个具有“blade”配置的服务器实例上。垂直伸缩指的是通过升级到更有伸缩性的硬件和操作系统(HW/OS)平台以利用多处理器配置的好处。一定要考虑到今天和将来对这两种伸缩性的要求。WebSphere Application Server、WebSphere Portal 和 Lotus Domino 都利用了垂直伸缩的好处，因为它们支持范围广泛的 HW/OS 平台，从低端的 Windows 服务器到高端的 IBM@server zSeries® 服务器，并可扩展到这些系统的多处理器配置。

应用程序水平伸缩能力的关键是在应用程序中维护的用户状态相关信息的数量。几乎无状态的应用程序比有大量用户/会话数据的应用程序水平伸缩要容易得多。在默认情况下，Domino 在使用一个应用程序期间要打开大量用户和会话信息，而基于 WebSphere 平台的应用程序是由应用程序开发人员决定他们所要管理和跟踪的信息。因此，不要求使用很多用户和会话信息的、基于 WebSphere 平台的应用程序可以比 Domino 应用程序更好地利用水平伸缩。

数据要求

数据模型

应用程序需要一个结构化的还是非结构化的数据模型?

有几种数据存储类型，不过，本文讨论的是关系和层次化。

关系存储维护“相关的”记录之间的多维链接。层次化存储使用一种相对扁平的父/子关系模型。通常，关系存储更适合于在记录中存在多对多关系的结构数据，如客户记录、发票、采购订单和厂商记录之间的联系。

层次化的存储更适合于非结构化的数据，如文档、列表、Lotus Freelance Graphics? 或者 PowerPoint 演示文稿以及其他形式的“自由格式”信息。层次化数据的另一个特点是改变数据结构的能力，如随意添加或者删除字段，而无需重新构建数据库。非结构化存储的其他特点有:

- *字段可以有多个值*
- *不同的应用程序或者客户可以同时编辑记录(没有记录锁定)*
- *最适合于相对简单(复杂性低)的搜索和文档内容搜索*
- *最适合于在目录清单或者一个表中显示的数据*

许多应用程序同时需要关系和层次化的数据存储，这也许是最适合创建集成 Lotus Domino 和 WebSphere Application Server 的功能的应用程序的情况了。Lotus Domino 有内置的非结构化数据存储，而 WebSphere Application Server 和 WebSphere Portal 有与 RDB 系统的健壮连接。还要记住的重要一点是层次化的 Domino 数据可以通过所提供的 portlet 经由 WebSphere Portal 表现，通过 IBM Lotus Domino Toolkit Studio 和 Domino Objects for Java 提供给 WebSphere 应用程序。

Portal 文档管理功能还让用户可以在 portal 中创建非结构化的数据，可以通过所提供的 API 访问它们。

应用程序是否需要访问记录系统?

记录系统是一种关系数据库功能，它强制实现记录的一个实例，并且只能有一个实例。这是许多行业应用程序的要求，如保险、健康和医药。尽管这是一个数据存储本身而不是应用程序平台的功能，但是 Domino 特有的包含自己的数据存储和复制模型的特点与记录系统正好相反。需要记录系统的应用程序一般依赖于关系数据存储，并且更常见于 WebSphere 平台。

数据量

数据量是否超过 2GB?

应用程序所需要的数据存储数量是决定其体系结构的重要因素。也要考虑其他数据相关的因素，如用户用创建/读取/更新/删除(CRUD)行动操作数据的程度和搜索要求。Domino Release 5 及以后的版本对 Domino 数据库的大小没有硬性的限制，但是一个关系数据库和 WebSphere 应用程序更适合于管理大数据量和频繁的 CRUD 操作。

不过，一个常见的例外是储存像文档这种非结构化数据和支持 Rich 文本或者有丰富格式的内容的需要，这些是 Lotus Domino 所擅长的。在这种情况下，理想的解决方案是构建一个集成的应用程序。如果需要在一个数据库中储存应用程序数据，那么 Lotus Enterprise Integrator (LEI)可以使 Domino 数据库与 RDB 同步。

同时，Domino 7 将支持储存在 DB2 中固有的信息以及 Notes Storage Facility。使用 DB2 可以为 Domino 数据提供更多的伸缩性和灵活性并使它可以集成到其他系统中。

数据分布

应用程序数据(不包括应用程序逻辑)是否需要分布到 WAN 或者 Internet 中的各个服务器上?