*分布式工作流系统的可扩展性和柔性研究

陶 冶　范玉顺　罗海滨
清华大学自动化系，北京 100084

摘　要　工作流管理系统为企业中不同部门之间的协作以及不同应用之间的集成提供了一个功能强大的计算环境。在当前工作流技术的研究领域中，分布式工作流系统逐渐成为研究的热点，而系统的可扩展性和柔性是分布式工作流系统成功实施的两个重要因素。本文介绍了相关的领域概念、系统结构、过程实例的执行以及全局控制机制，提出了一种基于企业功能领域划分的、具有高度可扩展性和柔性的分布式工作流系统的设计和实施方案。

关键字　工作流；分布式工作流系统；可扩展性；柔性；CORBA

一、 引言

工作流技术作为现代企业实现过程管理与过程控制的一项关键技术，为企业的经营过程提供了一个从模型分析、建立、管理、仿真到运行的完整框架。同时，工作流管理系统通过一套集成化、可互操作的软件工具为这个框架提供了全过程的支持。经过十几年的发展，工作流技术已逐渐走向成熟，至今已被应用于银行、电信、医疗保健、生产制造等诸多领域。

但是，也必须看到，工作流技术无论在理论上还是在技术本身还不够成熟：在经营过程中实际采用工作流管理系统的企业仍只是很少一部分，而且这些系统的应用范围也很有限，并不能全方位地支持企业的关键业务流程；此外，它的不足还表现在[1]：各工作流系统间彼此几乎都不兼容，无法胜任大规模业务，负荷能力有限，系统可扩展性、可靠性和灵活性差，而且不能提供强有力的安全保证，这主要是因为单一数据库、集中式结构、有限的通讯能力以及缺乏前瞻性的设计等原因所致。

工作流系统的分布程度对系统的整体性能有较大的影响，多工作流机的协作将使系统从结构上变得更加合理，它是解决工作流管理系统可扩展性和柔性的关键。本文将介绍一种基于企业功能领域划分的分布式工作流系统的协作模式和实施方法，它能够很好地解决以往集中式工作流系统所固有的一系列问题。

二、 企业的功能领域

在企业 CIMS 应用工程的实施过程中，需要首先根据企业自身特点，利用通用功能视图模型的基本构件生成对企业需求分析阶段的功能模型。在 CIMOSA 体系结构中[4]，功能视图反映了企业的功能结构，从上到下，依次为领域（Domain）、领域过程（Domain Process）、经营过程（Business Process）、企业活动（Enterprise Activity）和原子级的功能操作（Function Operation），其中的领域是一个很重要的概念。根据企业的经营目标，企业可以划分为若干个互相正交的领域，每个领域都有其目标、功能及其领域过程，领域的功能由领域过程实现。领域的划分和确定由企业经营目标和所受到的限制条件的集合决定。不同的领域将实现不同的企业目标，彼此之间通过事件、消息相互联系与协调，从而为企业建立新型的运行机制奠定了基础。

根据过程视图与功能视图之间的交互特性以及企业功能领域划分的思想，我们将把工作流管理系统中的核心组件工作流机与企业的建模过程有机结合，使运行时的系统能够与

* 863/CIMS 主题资助项目，课题编号：863-511-944-002

企业模型紧密集成。为此，需要从两个方面进行新的设置：首先，在建立每一个可执行的过程模型时，针对于最基本的活动单元，将它们与适当的功能领域相关联，这可以通过在建模工具中添加新的属性项来实现，有关功能领域的信息则由企业的功能模型提供；其次，对于每一个运行时的工作流机，也同样为它们绑定不同的功能领域，不同领域的工作流机负责执行不同的活动，这些设置在注册新的工作流机时完成。通过这种针对功能领域的绑定，企业中的每一个任务都将与某一个分布的工作流机发生联系。

三、 分布式工作流机的体系结构

按照 WfMC 提出的工作流参考模型[2]，一个工作流系统包括过程定义工具、工作流机、工作流管理工具、工作流客户应用和工作流机直接调用的应用等功能模块。同时，由于在实际应用中，工作流系统往往在不同的硬件平台、操作系统、网络协议、数据库的异构环境下运行，这对参考模型中各模块间的相互通讯和协作提出了很高的要求，CORBA 规范保证了这些要求得以实现，为系统运行提供了一个软件平台。

基于上面的考虑，本文提出如下介绍的系统体系结构：

整个系统是由过程建模工具、模型仿真工具、一个总控工作流机、一个工作表管理器和多个执行工作流机组成，其中的工作表管理器实际上是一个面向用户的执行工作流机。

就执行机、工作表管理器和总控机的关系来说，它们构成了一个 CORBA 平台上的圆锥形模型，总控机位于锥顶，各执行机位于锥底的圆周上，工作表管理器位于锥底的圆心上。总控机能监控工作表管理器和任何一个执行机，工作表管理器和执行机向总控机报告自己的状态，工作表管理器记录和维护各执行机所执行的任务。从系统级别上说，各执行机彼此之间的地位完全对等，只要需要，它们任何两两之间都可以建立联系。它们之间的关系可以形象地用图 1 来表示：

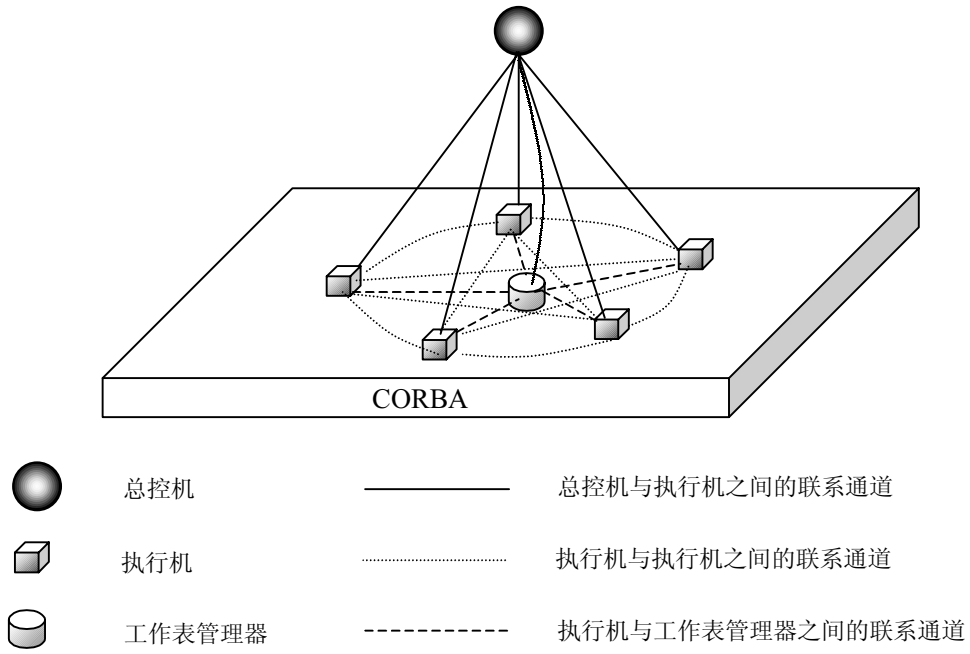

图 1　系统模型

建模工具负责创建工作流模型。在建模时，应对模型中的各个活动都绑定相应的功能领域。同时，各活动的输入输出、活动间转移的条件、非相邻活动间的数据连接都已事先定义。

总控工作流机与存储过程模型的中央数据库位于同一台机器上，主要负责指导建模工具进行模型分片、维护系统中各执行工作流机的配置信息、创建并激活过程实例、监控实例的运行状态等。执行工作流机则负责生成活动实例、激活应用、在活动之间进行导航、与工作表管理器交互等。工作表管理器则主要负责人工型活动任务项记录的生成、任务状态的更新与维护，这需要用户的参与。

整个系统采用基于 CORBA 规范的 Orbix Daemon 和 Orbix for Web 软件总线及其服务为系统支撑环境，系统提供的所有服务都经过 IDL 的定义与封装，经编译后以 CORBA 对象的形式出现。系统中所有的用户界面、管理员界面都是基于 Web 的。图 2 给出了系统的体系结构，并演示了一个简单过程实例的运行流程。

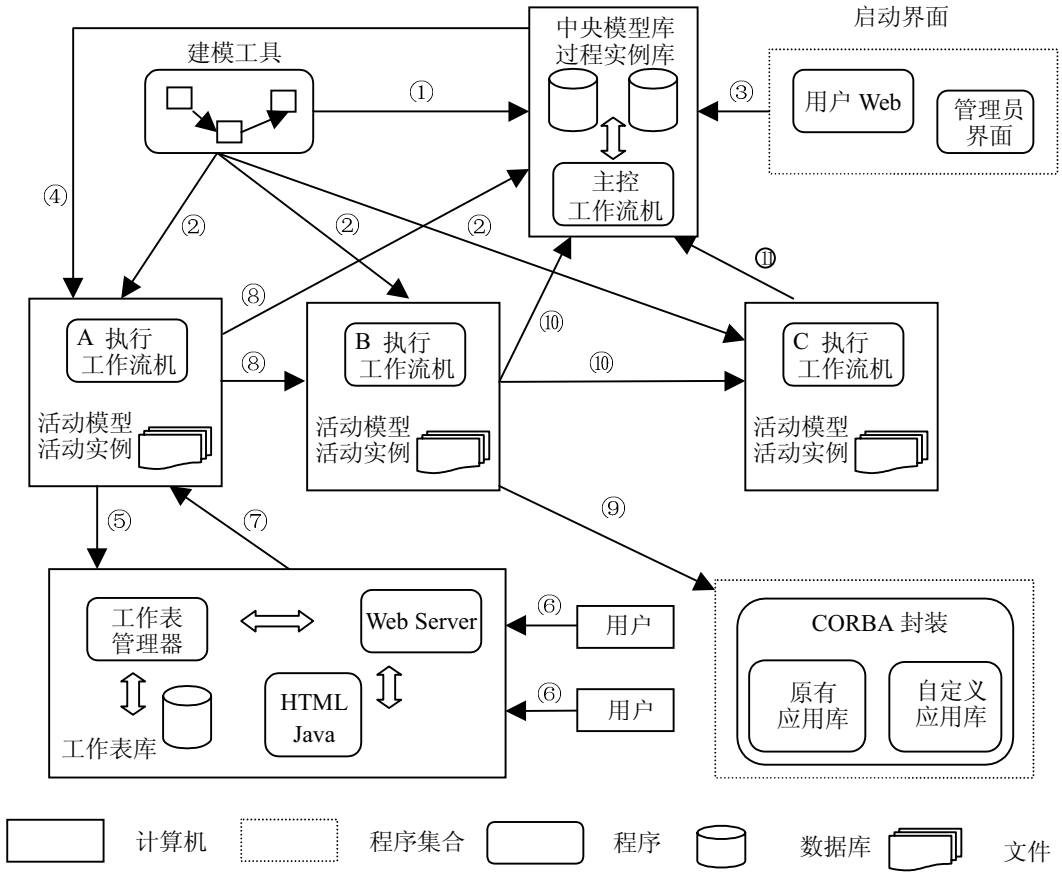

图 2　系统运行实例

具体的运行步骤解释如下：

（1）通过建模工具，一个由三个活动组成的过程模型被建立，并被保存到中央数据库中；

（2）过程模型被分片传递到三个领域的工作流机上；

（3）该过程被某一启动界面启动，首先由总控工作流机生成一个过程实例，然后通过查找数据库中的模型信息及有关工作流机的配置信息，获得负责执行第一个活动的工作流机的主机地址；

（4）调用工作流机 A 的接口函数，由 A 生成新的活动实例；

（5）该活动是一个人工型活动，因此，A 将激活工作表管理器在数据库中生成相应的工作项，工作项的分配方式是采取用户自主的 PULL 类型；

（6）用户访问自己的工作表，并执行任务；任务完成后，用户通过同样的界面进行提交，系统更新工作表；

（7）工作表管理器通知执行工作流机 A，该活动实例已经完成；
（8）工作流机 A 通过本地的模型分片文件及配置信息，通知工作流机 B 执行过程实例中的第二个活动，同时通知主控工作流机它已经完成第一个活动；
（9）执行工作流机 B 调用其他应用程序来完成第二个活动，这是一个 CORBA 封装的自动应用；
（10）类似地，工作流机 B 通知工作流机 C 来执行第三个活动，同时通知主控工作流机它已经完成第二个活动；
（11）C 通知主控工作流机，该过程实例已经执行完毕。

图 2 中 CORBA 封装的应用包括系统自带的原有应用库和用户自定义的应用库，这些应用都是一些诸如打印文档、发送邮件、定时等功能固定、可重用度较大的应用。这给用户提供了一个开放的接口，用户可以按系统提供的规范接口编写或设定他们自己的应用。这些应用以软件组件的形式出现，可以对之进行扩充和修改，从而可以提高系统的柔性和可重用性，便于企业信息系统的不断改进和重组。

四、 系统的可扩展性

分布式工作流系统相对于传统的集中式工作流系统而言，其优势在于分布机制可以有效地克服集中式工作流系统带来的瓶颈，使系统具有良好的可扩展性。

系统的可扩展性分为系统结构的可扩展性和系统功能的可扩展性。结构的可扩展性是指系统运行的硬件平台能根据需要方便地进行规模的扩充和压缩，而功能的可扩展性是指系统所具备的功能或能提供的服务能够根据需要进行添加或删除。由结构功能相关性原则，功能不能脱离结构而存在，要想获得功能上的可扩展性，必须先保证结构上具有相应的可扩展性。

从系统分析的角度来说，任何系统都包括一些关键功能和确保这些功能正常运行的关键资源，这些关键功能和关键资源统称为系统的关键组成要素[5]。关键组成要素集合包含的节点元素越多，则系统各部分之间的结合越紧密，关联越复杂，从而使系统的扩展越困难，可以说系统的可扩展性随着系统复杂度的提高而降低。因此，要提高系统的可扩展性，首先要从设计上降低系统组成结构的复杂度，其次要考虑降低系统中各元素之间的关联复杂度。

综合以上两点考虑，本系统从提高系统结构的可扩展性和系统功能的可扩展性两方面着手，通过尽量降低各工作流机之间的耦合强度，尽可能使分布式工作流系统在运行时能进行系统结构的扩展和功能的扩充。下面是相应的策略：

a. 使不同的执行机之间以及执行机与总控机之间能在线进行注册和注销

在系统中，总控机、执行机以及执行机提供的服务都通过 Orbix 提供的 Orbix Server Manager 进行系统注册，同时各服务对象对外提供透明的 IDL 接口。

以下是总控机和执行机各自与注册和注销相关的 IDL 接口：

```
struct EngineProperty
{
    string ip;                      //工作流机所在计算机的 IP 地址
    string name;                    //工作流机的名称
    string type;                    //工作流机的类型
    string domain;                  //工作流机所属的领域
    string current_start_time;      //工作流机本次启动时间
    string last_shutdown_time;      //工作流机上次关闭时间
```

```
        unsigned short serial_no;          //工作流机序列号
    };
    typedef sequence<EngineProperty> EnginePropertyList;

    interface WfControlEngine              //总控机接口
    {
        //注册新执行机
        boolean RegisterNewEngine(in EngineProperty new_engine, out unsigned short
serial_no, out EnginePropertyList other_engines, out unsigned short howMany);
        //关闭执行机
        boolean ShutDownEngine(in unsigned short serial_no);
        //获取执行机信息
        boolean GetEngineInfo(out EnginePropertyList engines, out unsigned short
howMany);
        //总控机提供的其它服务
        ……
    };

    interface WfEXEEngine                  //执行机接口
    {
        //更新配置文件
        boolean UpdateConfig(in EngineProperty own_property, in boolean flag);
        //执行机提供的其它服务
        ……
    };
```

一旦系统启动后，总控机在整个系统中主要是负责维护整个系统的配置信息，它向各执行机提供注册新执行机（RegisterNewEngine）、关闭执行机（ShutDownEngine）、获取执行机信息（GetEngineInfo）这三个相关的服务。每添加一个新的执行机或执行机的每一次启动都将调用总控机的 RegisterNewEngine 服务，此服务将向执行机返回其它已经注册且处于运行状态的执行机的配置信息。类似地，每关闭一个执行机，它都将调用 ShutDownEngine 服务，总控机将删除对应的执行机的记录。依靠这种方式，总控机在系统运行期间将一直保存着所有处于运行状态的执行机的配置信息。

执行机作为服务端可向其它执行机提供更新配置文件（UpdateConfig）的服务。比如每注册启动或关闭一个执行机 A，则执行机 A 将调用其它所有处于运行状态的执行机的 UpdateConfig 服务，以将执行机 A 的各项配置信息提供给它们，并将这些信息以配置文件的形式保存在各执行机本地。

图 3 是执行机 A 启动和关闭的简单的场景图：

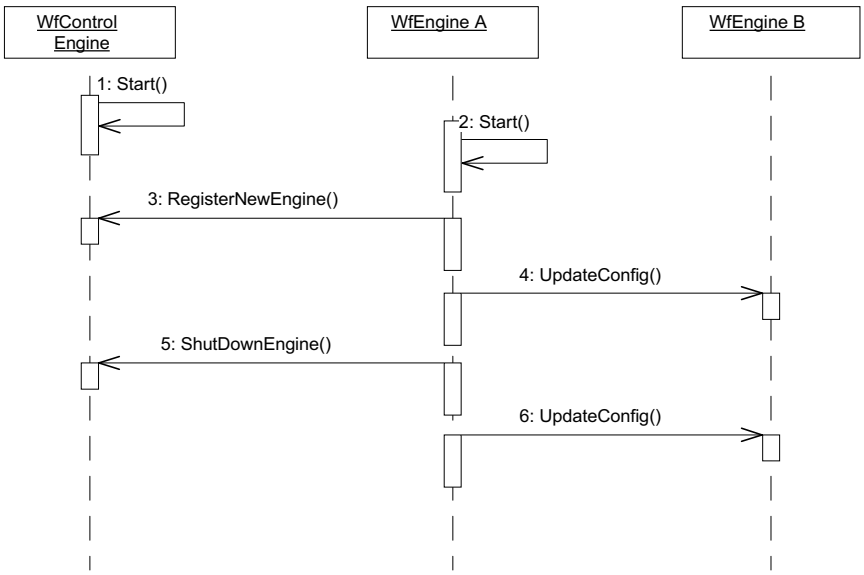

图 3　执行机启动和关闭场景图

从上面的 IDL 接口和执行机启动流程可以看出，各工作流机之间的关联仅在均处于运行状态的工作流机之间才存在，一旦某工作流机关闭之后，则它与整个系统的关联也就随之结束了。即使工作流机彼此之间存在关联，这些关联也只是每一个工作流机被动地接受其它工作流机的配置信息，除了总控工作流机和工作表管理器之外，各执行机之间的地位也是完全平等的，不存在任何主次之分，不存在执行机相互之间的控制、支配和制约。这样就避免了使某一工作流机成为系统显著的关键组成要素，降低了系统各组成要素之间的耦合强度，从而提高了系统的可扩展性。

对于总控工作流机和工作表管理器，由于它们在系统中所起的特殊作用和所处的特殊地位，它们的稳定性将影响系统的稳定性。增强总控工作流机和工作表管理器的稳定性可从两方面入手：一是增强程序的容错能力，二是采用双机冗余备份的策略。由于这两种工作流机的数量都只有一个，所以系统的成本增加并不是很大。若实在条件不允许，也可以只备份其中相关的数据库，一般的计算机即可以胜任。

有了上面总控机和执行机提供的这些服务，可以保证执行机相互之间以及执行机与总控机之间能在线进行动态注册和注销，而不会对系统运行的稳定性产生任何影响，这大大增强了系统的可扩展性。

b.　将执行机与之提供的服务进行动态配置

工作流机彼此之间的在线注册和注销仅仅从结构级上提供了一个可扩展的机制。但如果每一个执行机所提供的服务都是预先设定好的，用户无法修改，则每向系统中添加一项新的服务都要添加新的执行机，这意味着系统功能的扩展必然导致系统规模和复杂程度的膨胀，从而增加了整个系统的开销和复杂度，同时也就增加了监控管理的难度。因此，还应该从功能级上设法增强系统的可扩展性。将执行机与它所提供的服务进行分离，使服务能随时动态地配置到任何一个执行机上去，这是从功能级上提高系统可扩展性的一个重要机制。

在系统中，每个执行机都配置在一台计算机上，因此这台计算机里就配置了一个可向建模工具、总控机、工作表管理器和用户提供服务的 CORBA 对象。同时，系统中的原有应用或用户自定义的应用，比如打印服务、传真服务、定时服务、电子邮件服务等等，都通过 CORBA 封装成一个个独立的服务实体。这些实体也通过 Orbix Server Manager 进行注册，动态配置到执行机所在的计算机里。从本质上看，虽然执行机和上面这些应用都是系

统提供的服务对象，但可以认为这些应用是外挂在执行机下的，因为在系统运行中导航到某台执行机后，用户就能获得这台执行机所能提供服务的一个清单。这里实际上已将执行机的概念虚化了，因为它本身并不提供某种具体的面向用户或者面向业务的服务，它起到的作用就是对系统所提供服务的检索和管理，同时保证系统的正确导航。

有了上面这种机制的保障，在系统内部就可以自由地实现服务的动态配置，这样当系统功能需要扩充时，只需根据一定的规则将相应的服务随时外挂到某一个或几个执行机下，而无需添置新的计算机。这样既降低了系统的开销，也降低了对系统进行监控管理的难度，有效地提高了系统的可扩展性。

五、 系统的柔性

对制造系统和事务处理系统来说，系统的柔性同样是衡量系统性能的一个重要指标。系统柔性简单地说就是系统适应新情况、新环境、新任务的能力。一般地，柔性体现为环境、条件或过程状态发生变化时系统快速响应和跟随变化的能力。

一个系统的柔性可以从时间和范围两个方面来评价[6]。在时间方面，系统的柔性可以从系统达到某一适应程度所需的时间长度和在限定的时间内系统所能达到的适应程度这两点上来分别评价，这体现为系统的响应时间和系统的效率。在范围方面，系统的柔性可以从响应可预见变化和不可预见变化的能力这两方面来分别评价，这体现为系统的多功能性和系统的鲁棒性。

本文提出的分布式工作流系统，由于可以很方便地实现服务或功能的动态添加和配置，实现自动应用的添加、修改和删除，所以，当面临新的任务时，管理员或用户可以很快地调整系统的局部甚至整体的功能，使系统的响应时间很短，在新环境下实施新措施的效率很高。同样，若由于客观环境的不可预见的变化而要求系统结构马上做出相应的调整时，由于系统具备工作流机在线动态配置的功能，所以可以很快地实现这一目标。至于可预见的变化，系统更可以通过在建模工具中事先定义或通过建立策略库来快速响应甚至提前为这些变化做准备。这增强了系统的鲁棒性，同时使系统具备了实现多种不同的功能的潜力。

考虑一种极端情况，本系统现在正运行一套工作流，这时环境和工作任务都发生了变化，要求其中的一台执行机停止运行，而在另一个距离较远的地方新增一台执行机，同时在当前工作流中删除某一任务后再添加一项新的任务，还需要修改某一任务的参数，这可以通过注销工作流机、注册新的工作流机、在建模工具中进行任务增删和修改、在对应领域的执行机上配置新的功能或外挂新的自动应用、重新生成工作流模型实例、启动工作流等一系列步骤来实现，这体现了系统很高的柔性。

六、 结论

本文所提出的分布式工作流机的方案，按照对企业功能模型的需求分析与设计说明，工作流管理系统将在过程定义与执行这两个层次上对企业的功能领域提供支持。在定义过程时，将最基本的活动单元与适当的功能领域相关联；在执行过程时，不同的活动将由不同领域的工作流机来执行与导航。同时，系统采取了多种措施来提高系统的可扩展性和柔性。这种工作流机的分布式方案不仅可以解决原有集中式工作流管理系统所存在的一些瓶颈，而且能够与企业的建模过程紧密集成，从而形成一个完整的、具有良好可扩展性和柔性的、从建模到实施的过程体系，能较好地满足实际生产和业务流程的要求。

参考文献

1 范玉顺，吴澄，“工作流管理技术研究与产品现状及发展趋势”， <<计算机集成制造系统 CIMS>>，Vol.6，No.1，Jan.，2000，pp.1-7

2 Haibin Luo, Yushun Fan, “CIMFlow: A Workflow Management System Based on Integration Platform Environment”,7th IEEE International Conference on Emerging Technologies and Factory Automation,1999，pp.233-241

3 Alonso G, Agrawal D, Abbadi El A, et al. Functionality and Limitations of Current Workflow Management Systems. IEEE Expert, 1997,12(5).

4 王景光，甘仞初. “信息系统结构可扩展性分析”，System Engineering and Electronics，Vol 21，No.9，p37~39，1999

5 WfMC. Workflow Management Coalition Terminology and Glossary (WfMC-TC-1011). Technical Report, Workflow Management Coalition, Brussels, 1996.

6 William Golden, Philip Powell. Towards a definition of flexibility: in search of the Holy Grail? Oemga 28(2000) - The International Journal of Management Science. P373-384

Improving the Scalability and Flexibility of a Distributed Workflow System

Tao Ye, Fan Yushun, Luo Haibin
Department of Automation，Tsinghua University，Beijing 100084

Abstract Workflow management system (WfMS) has been used to provide a powerful computation environment for the integration of different applications and the cooperation of different organizations within enterprises. In the current research field of workflow technologies, the distributed workflow engines have gradually caught more and more attentions, and the scalability and flexibility of the whole system are the two critical factors to the successful implementation of the distributed workflow system. In this paper, a distributed workflow system with high scalability and high flexibility that runs on different enterprise function domains is introduced. Meanwhile, the relevant concept of domain, the system infrastructure, the implementation of a process instance and the global control mechanism are also presented.

Key words Workflow；Distributed Workflow Engine；Scalability；Flexibility；CORBA