

CIAgent 平台上基于 MAS 的企业分布式数据库集成应用研究

杨 力¹, 曹谢东¹, 陈毅红², 袁海燕¹

(1. 西南石油大学 计算机科学学院, 四川 成都 610500; 2. 西南民族大学 计算机科学学院, 四川 成都 610041)

摘 要: 为了克服传统多数据库集成技术的诸多不足, 提出了利用多 Agent 技术集成分布式数据库的新方法. 结合分布式人工智能中的多代理系统(MAS)与分布式数据库技术, 集成一个企业的多个现有数据库而成为一个新的同构异质分布式数据库, 在不影响原来的数据库应用的条件下, 提供全局模式, 在 CIAgent 平台上构建网络通信模块, 设计了多 Agent 系统结构, 实现多 Agent 交互. 运用该方法构建的油田分布式数据库系统能克服传统集成技术的许多不足, 具有良好的开放性、适应性和互操作性. 实践表明, 该方法为分布式数据库的集成提供了新的组织方式, 为油田的信息集成提供一种科学的新手段.

关键词: 企业信息集成; 分布式数据库; 多 Agent; CIAgent; 分布式事务; DIAgent

中图分类号: TP18, TP311. 133. 1 **文献标识码:** A **文章编号:** 1673-4785(2007)03-0073-06

Application research on enterprise distributed database integration using MAS with CIAgents

YANG Li¹, CAO Xie-dong¹, CHEN Yi-hong², YUAN Hai-yan¹

(1. School of Computer Science, Southwest Petroleum University, Chengdu 610500, China; 2. Computer Science Institute, Southwest Nation University, Chengdu 610041, China)

Abstract : To overcome many of the deficiencies of current multi - database integration technologies , this paper presents a new method for integrating distributed databases with Multi-agent-system (MAS) technology. Using MAS with heterogeneous databases allows the integration of multiple existing databases in an enterprise into a new distributed database with great heterogeneity , yet a global database platform is present without changing former database applications. This novel method of information integration based on MAS was applied to the management of oil - field data and network communication modules using a CIAgent platform. Workflow of MAS in the distributed database is discussed in detail , and the distributed database system was found to show good openness , adaptability , and interoperability. These results show the method provides a new way to organize the integration of distributed databases.

Key words : enterprise information integration ; distributing database ; MAS ; CIAgent platform ; DIAgent

到目前为止出现了一系列信息集成技术,从总体上可以分为 2 类:静态方法和传统的动态方法.其中静态方法中包括数据转移、通用 SQL API、通用数据访问结构 ODBC 和数据库网关等,他们通常是点对点的集成,当信息系统增多时,其问题的复杂性成指数级增长;传统的动态方法包括 DCOM/

CORBA/JAVA BEAN 等分布式中间件集成,一般采用模块式结构,实现起来比较复杂,修改维护不便,并且不同的中间件难于实现互操作^[1].为此,通过分析分布式数据库技术与多 Agent 技术,利用 Agent 的智能性、移动性、跨平台性及交互性,提出了用多 Agent 技术动态方式集成现有数据库系统为分布式数据库,同时支持在此基础上的全局应用,结合油气田信息系统的实际,构建了一个基于 CIAgent 平台的 MAS(多代理系统)信息集成系统,为

收稿日期:2006-07-07.
基金项目:国家高技术研究发展计划(863)资助项目(2003AA414030).

油气田信息集成提供了一种新的方法。

1 多 Agent 与分布式数据库相结合的优势

多 Agent 技术与分布式数据库管理技术结合有如下优势。

1.1 智能相似性

运用多 Agent 技术可以充分发掘分布式数据库管理中的智能性。要实现分布式数据库,必须建立数据字典。字典中的一些内容,如:数据的分片、分配、角色等信息,均可视为规则知识;另外,查询优化算法中的优化规则也是一种规则知识,这与人工智能的知识表示是类似的,对这些规则的使用,就是推理。传统的分布式数据库管理系统对规则的管理和使用,没有智能性,存在一定局限性。而文中提出的采用多 Agent 技术是分布式人工智能技术的一种,充分体现了对规则进行管理使用的智能性,为数据库管理系统技术开辟了新的视野。

1.2 复杂性一致

利用多 Agent 技术可以解决分布式数据库管理中的复杂性问题。对于涉及由大量具有主动性的个体组成的复杂系统,如经济社会系统、交通系统等,由于其动态性、随机性很强,很难仅仅依赖数据数学模型进行仿真。由于分布式数据库系统的分布性,造成了系统固有的复杂性,采用传统的模块化方法,无法完全满足分布式数据库技术的要求,考虑实现的代价,目前能够完全实现分布式功能的商业化产品几乎没有,只是部分实现,如果用多 Agent 技术,则可以在一定程度克服这些问题。

1.3 系统结构类似性

多 Agent 系统结构与分布式数据库管理系统之间具有结构一致。多 Agent 系统是由多自治的 Agent 组成的,各 Agent 之间相互协作;分布式数据库系统中是由多个局部数据库系统组成的,各个局部数据库系统之间为了实现分布式查询、事务处理、并发控制等需要相互协作。由此可以看出它们的相似性,所以应用多 Agent 技术实现分布式数据库具有天然优势。

1.4 并行性一致

多 Agent 系统与分布式数据库管理系统之间具有并行性一致的特点。多 Agent 系统中各 Agent 之间具有高度的并行性,而分布式数据库系统中各局部数据库系统也是并行,二者相类似。

1.5 同源于分布式技术

分布式数据库技术是分布式技术与数据库技术结合的产物。分布式技术与人工智能结合产生了分

布式人工智能,分布式人工智能分为分布式问题求解及多 Agent 技术。所以分布式数据库技术及多 Agent 技术都源于分布式技术,它们都是分布式技术与其他技术相结合产生的,故把这 2 种技术结合也是很自然的。

2 CIAgent 平台介绍

Joseph P. Bigus 和 Jennifer Bigus 研制了一个构建智能 Agent 环境 CIAgent (constructing intelligent agents)。CIAgent 全部采用 Java 语言编写,可以在 JDK1.1 环境下运行。CIAgent 智能主体环境具有下列特点:1) 很容易在已有的 Java 应用系统中加入主体;2) 提供生成主体的图形构建工具。JDK1.1 包含 Java. beans 软件包,具有可视建造工具,并且可以嵌套定义;3) 主体能支持相当强的事件处理能力;4) 可以将领域知识加到主体知识库中;5) 主体具有一定的学习功能;6) 多主体是永久的,可以随时将它写入文件或装入运行^[2]。

该开发工具的一个最大优点是源代码是开放的,通过深入分析源代码,确保在应用开发时正确使用开发包,同时为用户扩展它的功能提供了方便。同时笔者对本开发工具进行了扩充,增加了网络通信模块,解决了网络环境下多 Agent 之间的交互。另外,基于 Java 的开发包有利于集成异种平台上数据库系统。

3 CIAgent 平台上基于 MAS 的企业分布式数据库集成体系结构

3.1 基于 MAS 的企业信息集成框架的提出

以中石化川西采输处为试验基地,由于采输处的组织结构为机关总部管理着多个采气分队和输气分队,其组织结构为一种主从式层次结构,所以本系统的多 Agent 组织采用行政管理型^[3],其 DIAgent (distributed intelligence agent) 分为 3 类:1) 管理 DIAgent,具有所管理的 DIAgent 名字、地址、能力;2) 处理分布查询的查询分解 DIAgent;3) 处理分布事务的事务协调 DIAgent^[4]。设计 DIAgent 系统结构如图 1。表 1、2 是图 1 中的各个 DIAgent 的名称。

表 1 各采气队 DIAgent 表

Table 1 Gas production team DIAgent table

采气队名	物理查询 DIAgent 子	事务 DIAgent
采气 1 队	YDSubTransactionDIAgent	YDPhyQueryDIAgent
采气 2 队	SDSubTransactionDIAgent	SDPhyQueryDIAgent
采气 3 队	TDSUBTransactionDIAgent	TDPhyQueryDIAgent
采气 4 队	FDSUBTransactionDIAgent	FDPhyQueryDIAgent

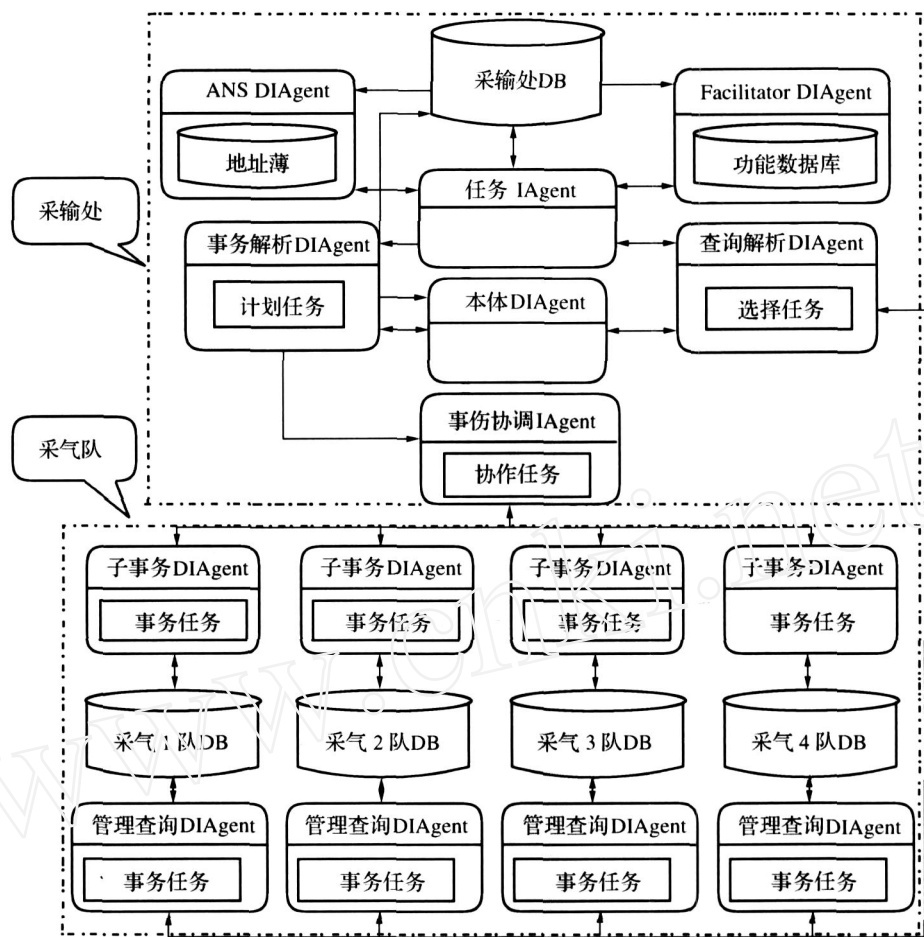


图 1 多 Agent 系统结构

Fig.1 MAS system structure

表 2 采输处各 DIAgent 的端口号表(CSCAgent Port)			
Table 2 Gas production team DIAgent Port table			
DIAgentName	IPAdress	port ID	rem
ANSDIAgent	192. 168. 2. 11	2501	名字解析服务 DIAgent
FacilitatorDIAgent	192. 168. 2. 11	2502	功能解析服务 DIAgent
OntologyDIAgent	192. 168. 2. 11	2503	本体 DIAgent
TaskAnalyse DIAgent	192. 168. 2. 11	2504	任务 DIAgent
TransactionPlan DIAgent	192. 168. 2. 11	2505	事务解析 DIAgent
QueryPlan DIAgent	192. 168. 2. 11	2506	查询解析 DIAgent
Transaction CorDIAgent	192. 168. 2. 11	2507	事务协调 DIAgent

3.2 DIAgent 交互过程

分布式查询或分布式事务处理时,多个 DIA-

gent 之间的交互过程分析如下(B-G 是分布式查询处理过程,H-N 是分布式事务处理过程). 协议对故障处理办法:采用超时方法处理. 当在规定的时间内没有收到回复,则认为失败.

1)任务 DIAgent 接收任务,判断是查询,还是事务,如果是查询,则把任务交查询解析 DIAgent,转 2);如是事务,则把任务交事务分解 DIAgent,转 8).

2)查询解析 DIAgent 分析该查询并按照一定的算法把分布查询分解为物理查询.

3)查询解析 DIAgent 按照物理查询的功能要求到功能解析服务 DIAgent (即 facilitator DIAgent) 查询哪一 DIAgent 能够完成此任务, facilitator DIAgent 找到后,则返回 DIAgent 名称. 在规定时间内,如不能解析则结束本次查询任务处理.

4)查询解析 DIAgent 利用 3) 步得到的 DIAgent 名称,从自己熟悉 DIAgent 列表中查询地址. 如找不到,则到名字解析 DIAgent 查询该 DIAgent 对应的地址 (包括 IP 地址和端口号). 在规定时间内

内,如不能解析则结束本次查询任务处理。

5) 查询解析 DIAgent 按地址向目的物理查询 DIAgent 发送查询处理消息,重复 3)、4)、5) 直接把复合查询的物理查询处理完成为止。在规定时间内,如果某一个物理查询任务不能提交到目的物理查询 DIAgent,则本次查询失败,结束本次查询任务处理。

6) 各物理查询 DIAgent 完成查询任务后,返回查询结果到查询解析 DIAgent。在规定时间内,如不能返回结果,则本次查询失败。

7) 查询解析 DIAgent 综合物理查询结果,提交任务 DIAgent,由任务 DIAgent 交给用户/应用系统,结束本次查询任务处理。如果某一个物理查询 DIAgent 不能返回查询结果,则结束本次查询任务处理。

8) 事务分解 DIAgent 分析接收的任务是不是分布式事务,如是分布式事务,则按照一定的算法把分布式事务分解为一系列子分布式事务。

9) 事务分解 DIAgent 按照子事务/事务的功能要求到功能解析服务 DIAgent (即 Facilitator DIAgent) 中查询哪一个 DIAgent 能够完成此任务, Facilitator DIAgent 找到后,则返回 DIAgent 名称。如不能解析,则结束本次事务处理。

10) 事务分解 DIAgent 利用 9) 步得到的 DIAgent 名称从自己熟悉 DIAgent 列表中查询地址。如找不到,则到名字解析服务 DIAgent 查询该 DIAgent 对应的地址 (包括 IP 地址和端口号)。如不能解析则结束本次事务处理。重复 9)、10) 直接把分布式事务的子事务按上述要求作完。

11) 事务分解 DIAgent 把上面的所有处理子事务的 DIAgent 的地址及子事务内容以消息方式告知事务协调 DIAgent。事务协调 DIAgent 把所有的子事务发送到处理子事务的目的 DIAgent (即各采气队的子事务 DIAgent)。

12) 事务协调 DIAgent 向所有相关的子事务 DIAgent 发送“开始提交”消息。由于每个相关的子事务 DIAgent 据自己的情况进行投票,只有收到所有的相关的子事务 DIAgent 均赞成,提交才进入第 2 阶段 13) 和第 3 阶段 14),如在规定时间内收不到回复,则认为参与者拒绝处理事务,结束本事务处理。

13) 事务协调 DIAgent 向所有相关的子事务 DIAgent 发送“准备提交”消息。相关的子事务 DIAgent 收到该报文后,若已经准备好提交,则回答“准备就绪”消息,否则结束本事务处理。

14) 当事务协调 DIAgent 收到所有的相关的子事务 DIAgent “准备就绪”回答,就向所有相关的子

事务 DIAgent 发“提交”报文,此时每个相关的子事务 DIAgent 知道其他的相关的子事务 DIAgent 将“赞成”提交,故它可以收到“提交”报文后就进行提交^[5-6]。

3.3 网络通信模块的设计

3.3.1 设计思想

网络分布环境中各 DIAgent 之间需要通信,是通过网络通信模块进行的,每个 DIAgent 都需要一个网络通信模块,设计网络通信模块要考虑以下几个因素:

1) 网络通信模块支持与外界双向通信,与外界的通信接口应该有收发 2 个端口。

2) 网络通信模块支持与宿主双向通信,所以与宿主的通信接口应该有 2 个端口。

3) 网络通信模块内应该具体缓冲区,保证通信的可靠性,防止因拥塞而掉包发生。

4) 网络通信模块作为服务方接收信息时,应支持多客户访问。

为此,根据每个通信的要求设计了下列网络通信模块图:本模块特点是各部分高度并行运行,保证了在有多数 Agent 的条件下通信要求,其结构如图 2。

3.3.2 网络通信模块组成

Socket 接口:功能是将直接与协议有关的通信部分组合在一起,并给网络通信模块的其他部分提供一种通信方式,使网络通信模块的其他部分不用考虑与发送协议有关的部分,各部分的工作原理如下:

服务线程:它使用 ServerSocket 不停地监听本 DIAgent 的端口地址,一旦发现消息来,主动启动一个消息线程处理这个消息,然后继续监听,它支持多客户。

消息线程:由服务线程启动。它的任务是读入消息,并将消息送到接收缓冲区。

客户线程:客户线程由发送线程启动,任务是将消息通过 Socket 发出。

接收缓冲区:用来缓存外界发来的消息,它的访问方式采用同步,即互斥访问。

发送缓冲区:用来缓存向外发送的消息,以免连续发送多个消息而使系统出错。向外发送的消息一般是通过调用网络通信模块的消息发送函数进入发送缓冲区。根据本模块情况,发送缓冲区和接收缓冲区按队列原理工作。

发送线程:是一个常驻线程,它的任务是不断监视发送缓冲区,一旦有消息进入发送缓冲区就启动 Socket 接口中的方法来将消息发出。Socket 接口中的方法将产生一个客户线程来将消息发出。

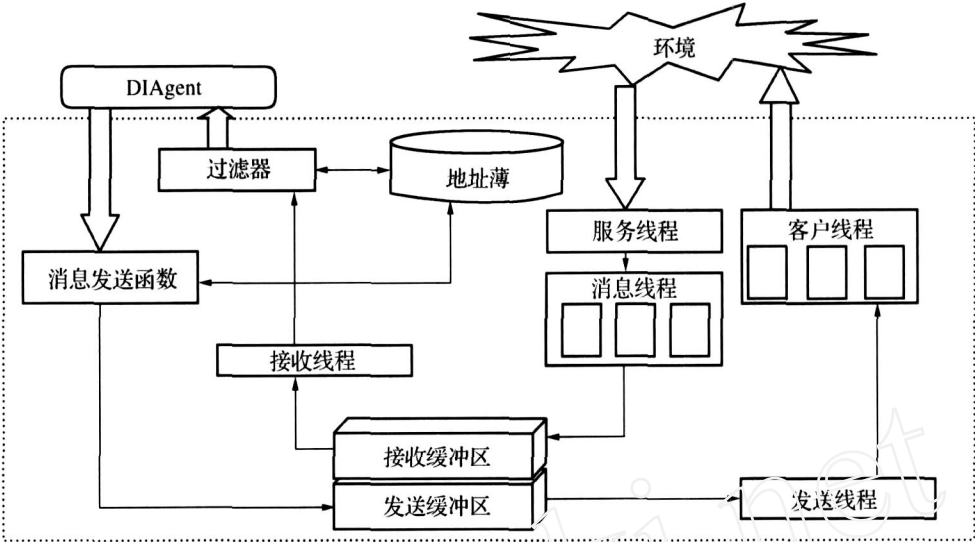


图 2 网络通信模块结构

Fig. 2 Network communication module structure

接收线程:是一个常驻线程,任务是不断检查接收缓冲区的数据,一旦有消息进入接收缓冲区就启动一个过滤线程来接受消息进行解释和处理。

过滤器:过滤器解释并处理一些简单的且仅与网络通信模块有关的一些通信原语。如查询地址或者告诉地址 register,unregister 等。网络通信模块的过滤器功能比较弱,大部分与 DIAgent 有关的消息还是交给 DIAgent 去处理,过滤器对 ANSDIAgent 特别有用。

地址簿:网络通信模块保留 DIAgent 的地址信息。地址簿中包含 DIAgent 熟人的地址,比如:

QueryPlanDIAgent 保存有 PhyQueryDIAgent 地址,不需查询 ANSDIAgent,当然也可查询。对于通信服务器而言,地址簿保存有系统中所有已登记的 DIAgent 地址。

消息发送函数:网络通信模块的消息发送函数首先查阅地址簿中有没有目标 DIAgent 的地址,如果没有,先产生一个 Query_address 消息(增加的语用词),向通信服务器查询目标 DIAgent 的地址,然后将自己挂起一段时间,直到地址簿中出现目标 DIAgent 的地址。如果持续一段时间还是得不到地址,则发送失败。在知道了目标 DIAgent 的地址后,消息发送函数把消息存放在发送缓冲区。网络通信模块主要是按照上层提供的逻辑地址,将其转换为地下(消息接收则相反),并将 KQML 消息线型化(或反之),然后打包发送或解包。消息发送函数与发送线程之间均需要访问发送缓冲区,二者是生产者与消费者的关系,二者需进行同步。

消息接收实现原理:在 bean 类(CIAgent 是

bean 的子类)设置一个属性,当过滤器收到 KQML 消息后,就存放在属性中,于是就会产生一个属性改变事件,对应此事件,有一个监听者,在其中实现一系统的处理 KQML 消息函数(即所谓注册函数)。

4 结束语

MAS 系统是一种新分布式信息集成模式的核心技术。作为分布式数据库集成的一种方案,MAS 其实是对传统的静态和动态集成方法的一次革命。通过在中石化川西采输处的实施,把 MAS 技术应用到采输处各分队相对独立的子数据库的集成中,证明本系统具有良好的开放性、适应性和可操作性,通过引入了 DIAgent 对支持网络分布计算环境的支持,使得各计算机系统 DIAgent 实现问题协作处理;虽然它在企业专网上实现,但它的跨平台性、移动性和弱连接性决定了它同样适合 Internet 广域网,其本身所具有的优越性和强大生命力决定了它的发展前景必然十分广阔。

参考文献:

[1]GU P, BALASUBRAMANIAN S, NORRIE D H. Bidding-based process planning and scheduling in a multi-agent system[J]. Computers Industry Engineering, 1997, 32 (2) :477 - 496.

[2]HEECHEOL J, CHARLES P, MARK R C. A Java Agent infrastructure with message routing[A]. Stanford Center for Design Research (CDR) [C]. Stanford, USA, 1999.

[3]张 洁,高 亮,李培根.多 Agent 技术在先进制造中的应用[M]. 北京:科学出版社,2004.

[4]CARVER Z, CVETANOVIE Z, LESSER V. Sophisti-

- cated cooperation in FA/C distributed problem solving systems [A]. Proceedings of the 9th national conference on Artificial Intelligence[C]. Anaheim, CA, 1991.
- [5] CORA B EXCELENT TOLEDO. The dynamic selection of coordination mechanisms in Multi-agent Systems[D]. Southampton: University of Southampton, 2003.
- [6] CUTKOSY M R, TENENBAUM J M, BROWN D R. Working with multiple representations in a concurrent design system[J]. Journal of Mechanical Design, 1992, 114(3): 515 - 524.

作者简介:



杨力,男,1975年生,讲师,主要研究方向为人工智能与分布式计算,参加过国家“十五”科技攻关项目在内的多个大型项目,其中一项获得四川省科技进步二等奖,在计算机类核心期刊上发表论文多篇。

E-mail: scncyl @126.com.



曹谢东,男,1954年生,教授,硕士生导师,中国人工智能学会智能系统工程专业委员会副主任,主要研究方向为人工智能与分布式计算。完成国家、省、部级项目多项,其中2项获四川省科技进步三等奖。著专著3部,专利1项,在国内外学术会议和刊物发表论文60多篇。



陈毅红,男,1973年生,硕士,讲师,主要研究方向为人工智能与分布式计算。参加过多个实际工程项目,其中参与的气田安全整改项目获得四川省科技进步三等奖;在计算机类核心期刊上发表多篇论文。

计算智能及安全国际会议 2007 International Conference on Computational Intelligence and Security (CIS2007)

International Conference on Computational Intelligence and Security (CIS) is a major annual international conference to bring together researchers, engineers, developers and practitioners from academia and industry working in all areas of two crucial fields in information processing: computational intelligence (CI) and information security (IS), to share the experience, and exchange and cross-fertilize ideas. In particular, the series of CIS conference provides an ideal platform to explore the potential applications of CI models, algorithms and technologies to IS. Following the great success of CIS2005 and CIS2006, CIS2007 serves as a forum for the dissemination of state-of-the-art research, development, and implementations of systems, technologies and applications in these two broad fields. CIS2007 is co-organized by Harbin Institute of Technology and Guangdong University of Technology. Also, it is co-sponsored by Xidian University, IEEE (Hong Kong) Computational Intelligence Chapter, and Hong Kong Baptist University.

Prospective authors are invited to submit high-quality manuscripts written in English. The submission of a paper implies that the paper is original and has not been submitted to elsewhere for possible publication. All submissions will be blindly reviewed by experts in the field based on originality, significance, quality and clarity. Authors must not reveal any author information in the submitted paper(s), otherwise the paper(s) will be rejected immediately without going through the reviewing process. Authors should use the Latex style files or MS-Word templates provided by the IEEE to format their papers. The length of a submitted paper should not exceed 4 pages in two-column format. The accepted papers will be included in the conference proceedings published by the IEEE that is indexed by EI.

Authors should submit a soft copy of their manuscripts with the pdf format only to the conference via the online submission system. For more details, please visit the conference web site at: <http://cis2007.hit.edu.cn>