

## 量子程序设计语言 NDQJava 处理系统\*

宋方敏<sup>+</sup>, 钱士钧, 戴静安, 张云洁, 徐家福

(南京大学 计算机软件新技术国家重点实验室, 江苏 南京 210093)

### Processing System of Quantum Programming Language NDQJava

SONG Fang-Min<sup>+</sup>, QIAN Shi-Jun, DAI Jing-An, ZHANG Yun-Jie, XU Jia-Fu

(State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China)

+ Corresponding author: Phn: +86-25-83593673, Fax: +86-25-83593673, E-mail: fmsong@nju.edu.cn, <http://www.nju.edu.cn/>

Song FM, Qian SJ, Dai JA, Zhang YJ, Xu JF. Processing system of quantum programming language NDQJava. *Journal of Software*, 2008,19(1):9-16. <http://www.jos.org.cn/1000-9825/19/9.htm>

**Abstract:** This paper describes one of the processing systems for the quantum programming language NDQJava. Its main feature lies that the classical parts of NDQJava programs are processed by the Java processing system, so this effort emphasizes on the processing of program's quantum parts. This processing system follows the compilation-interpretation approach, and it includes lexical analyzer, syntactic analyzer, code transformer, quantum assembler and quantum interpreter. By the end of this paper, some examples are given. The system was implemented by simulation in June 2006 on the classical computer.

**Key words:** quantum programming language; compilation; interpretation; simulation; lexical analyzer; syntactic analyzer and code transformer; quantum assembler and quantum interpreter

**摘要:** 简要介绍了量子程序设计语言 NDQJava 的一个处理系统,其特点是:程序中经典部分之处理借助 Java 系统,着重考虑量子部分之处理,该处理系统遵循编译-解释的途径,由词法分析程序、语法分析与代码转换程序以及量子汇编与解释程序 3 部分组成,文末还给出了示例,该系统已于 2006 年 6 月底在经典计算机上模拟实现。

**关键词:** 量子程序设计语言;编译;解释;模拟;词法分析程序;语法分析与代码转换程序;量子汇编与解释程序

中图分类号: TP301 文献标识码: A

## 1 工作缘起

Shor 关于数的质因子分解算法<sup>[1]</sup>以及 Grover 关于数据库搜索算法<sup>[2]</sup>的相继出现,显示出量子计算机之功能有可能超越经典计算机.1996 年出现的 Knill 的量子伪码<sup>[3]</sup>、Baker 的 Qgol<sup>[4]</sup>、1998 年 Ömer 的 QCL<sup>[5]</sup>、2001 年 Zuliani 的 qGCL<sup>[6]</sup>以及 2005 年 Altenkirch 等人的 QML<sup>[7]</sup>等堪称量子程序设计语言这一领域的代表性工作.2005 年,我们在学习前人工作的基础上设计出一种基于混成式量子计算机结构的命令式量子程序设计语言 NDQJava.其经典成分采用 Java 语言,量子成分自行设计.主要包括量子数据类型 qtype、量子变量、量子表达式、

\* Supported by the Natural Science Foundation of Jiangsu Province of China under Grant No.BK2007138 (江苏省自然科学基金)

Received 2007-05-18; Accepted 2007-06-30

量子语句等.程序与 Java 程序类似.不同之处有二:一为量子变量说明,二为量子分程序.2006 年春着手考虑 NDQJava 的实现,决定遵循编译与解释相结合的原则.同年 6 月底,在本实验室的经典计算机上模拟实现了它的一个处理系统.

## 2 实现途径

语言处理系统一般采用 3 种实现途径,即编译、解释和先编译后解释.

编译途径指的是由编译程序对整个源程序进行分析,将它翻译成等价的目标程序,然后执行.其优点是只需分析和翻译源程序一次,一旦获得目标程序便可反复运行,而不必重复编译.不足之处在于,当目标程序在运行中发现错误时难以查找出错位置,并且为该编译付出的存储开销较大.

另一种途径是解释.与编译不同,解释程序并不产生目标程序,而是对程序逐个语句边分析边执行.其优点是易于查错,因为若存在错误,可以快速从正在执行的程序中确定出错语句的位置,这对程序调试很有帮助,并且为解释付出之存储开销较小,但是每当要执行一个源程序语句时,解释程序就要对其分析,以便发现如何完成所要执行的功能.显然速度慢,效率低.

第三种途径是先编译后解释.首先通过编译程序分析整个源程序,并将之翻译成中间语言程序,然后解释执行该中间语言程序.设计中间语言的目的是为了缩短执行每个语句所需的分析时间,以及便于在不同的平台上运行.

鉴于 NDQJava 语言由 Java 语言扩展而来,它以 Java 为其子语言,而一般 Java 处理系统采用了先编译后解释的途径.为便于程序中的经典部分委交 Java 系统处理,故而在 NDQJava 处理系统中也就采用了这种途径,先由编译程序将 NDQJava 程序编译成 Java 中间语言程序,然后由 Java 解释程序辅以量子汇编与解释程序解释执行.

## 3 设计准则

本系统的设计准则与 NDQJava 语言的设计准则相同,即实用性、简明性、严谨性以及快速性,详见文献[8].

## 4 设计难点

本处理系统的设计难点有二:一是量子成分的编译处理,二是量子汇编与解释程序和 Java 虚拟机接口处理.为使语言处理尽可能地借助 Java 系统,针对量子类型 `qtype`,用 Java 书写了一个与之相应的 Java 类 `QType`,在其中定义了处理量子变量所需的变量与操作.由于 Java 语言有外部过程,故而量子汇编与解释程序和 Java 虚拟机的接口可以以外部过程调用接口的形式置于 `QType` 类中.这样,便利用 Java 系统解决了量子成分的处理,从而矛盾转化成 Java 类 `QType` 以及量子汇编程序与解释程序的编写.

## 5 系统概貌

(1) 本系统由 3 部分组成,即词法分析程序、语法分析与代码转换程序以及量子汇编与解释程序,系统总框架如图 1 所示.

(2) 词法分析程序.词法分析程序的功能是切分 NDQJava 源程序的单词,识别出各个标识符的类型,给出单词属性字序列、单词类表和类型表,标记出标识符是经典类型还是量子类型.本词法分析程序由词法分析程序 1 以及词法分析程序 2 组成.前者负责切分 NDQJava 源程序的单词,后者负责识别标识符的类型,并判定是经典类型还是量子类型,其中,单词属性字的格式是:

|       |      |     |     |
|-------|------|-----|-----|
| CQ 标记 | 单词类号 | 单词号 | 类型号 |
|-------|------|-----|-----|

CQ 标记:0 表示经典,1 表示量子.

单词类号:0 表示常量,1 表示分界符,2 表示关键字,3 表示标识符.

单词号:单词在对应单词类表中的序号.

类型号:单词的类型在类型表中的序号.

单词类表有 4 种,即常量表、分界符表、关键字表以及标识符表.其统一格式为:

|     |    |
|-----|----|
| 单词号 | 单词 |
|-----|----|

单词号已如前述,单词是指单词在源程序中的一列字符.类型表的格式为:

|     |    |
|-----|----|
| 类型号 | 类型 |
|-----|----|

其中类型号的含义如前,类型为类型名.

词法分析程序的细节参见文献[8].

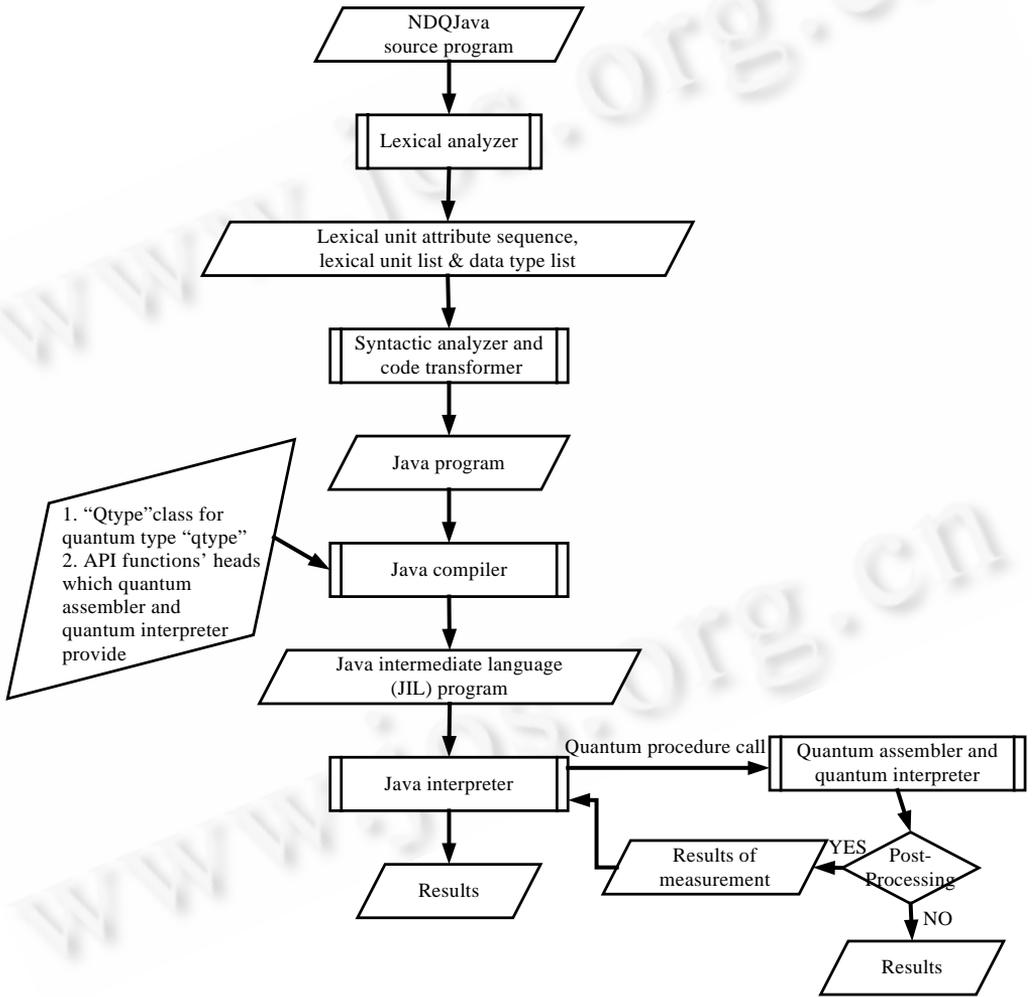


Fig.1 Flow diagram of the NDQJava processing system

图 1 NDQJava 处理系统框图

(3) 语法分析与代码转换程序.

语法分析与代码转换程序的功能是对单词属性字序列进行语法分析,将 NDQJava 程序中的经典部分还原,量子部分转换成相应的 Java 代码.

由于 Java 编译程序无法处理量子类型.如前所述,采用 Java 语言中的特定成分书写了一个与 qtype 相应的

QType 类,其类体组成如下:

- 一是为对量子变量分配存储而定义之有关变量;
- 二是相应各种酉操作、张量积操作、测量操作以及为实施量子变量存储分配所定义的函数;
- 三是为量子汇编与解释程序和 Java 虚拟机相接所需之外部过程调用接口。

在代码转换时,当遇到 `qtype` 类型的量子变量  $q$  之说明,便将 `qtype` 换成 QType,从而得到 QType  $q$ ,在分配存储时,即生成 QType 类的一个实例.当遇到 `qtype` 类型之操作时,便将之转换成 QType 类的相应函数调用.量子操作与 QType 类的成员函数对应表见表 1.

**Table 1** Mapping table of quantum operations to the member functions in QType class

**表 1** 量子操作与 QType 类成员函数对应表

| Quantum operations   | Operations on the datatype "qtype" | Member functions of the class "QType"                  |
|--|------------------------------------|--|
| Identity transformation on the $p$ -th qubit   | $u\_i(p)$                          | $u\_i(int p)$  |
| X transformation on the $p$ -th qubit  | $u\_x(p)$                          | $u\_x(int p)$  |
| H transformation on the $p$ -th qubit  | $u\_h(p)$                          | $u\_h(int p)$  |
| S transformation on the $p$ -th qubit  | $u\_s(p)$                          | $u\_s(int p)$  |
| Z transformation on the $p$ -th qubit  | $u\_z(p)$                          | $u\_z(int p)$  |
| Rotation transformation on the $p$ -th qubit about the X axis by $d$ rad.  | $u\_rotx(p,d)$                     | $u\_rotx(int p,double d)$                              |
| Rotation transformation on the $p$ -th qubit about the Y axis by $d$ rad.  | $u\_roty(p,d)$                     | $u\_roty(int p,double d)$                              |
| Rotation transformation on the $p$ -th qubit about the Z axis by $d$ rad.  | $u\_rotz(p,d)$                     | $u\_rotz(int p,double d)$                              |
| Controlled-Not transformation on the $p$ -th qubit (the $c$ -th qubit as controlling qubit)                                  | $u\_cnot(c,p)$                     | $u\_cnot(int c,int p)$                                 |
| Controlled-Rotation transformation on the $p$ -th qubit about the X axis by $d$ rad (the $s$ -th qubit as controlling qubit) | $u\_crotx(s,p,d)$                  | $u\_crotx(String s,int p,double d)$                    |
| Controlled-Rotation transformation on the $p$ -th qubit about the Y axis by $d$ rad (the $s$ -th qubit as controlling qubit) | $u\_croty(s,p,d)$                  | $u\_croty(String s,int p,double d)$                    |
| Controlled-Rotation transformation on the $p$ -th qubit about the Z axis by $d$ rad (the $s$ -th qubit as controlling qubit) | $u\_crotz(s,p,d)$                  | $u\_crotz(String s,int p,double d)$                    |
| Swap operation between $p_1$ -th and $p_2$ -th qubits  | $u\_swap(p_1,p_2)$                 | $u\_swap(int p_1,int p_2)$                             |
| Global phase transformation by $d$ rad (the $s$ -th qubit as controlling qubit)  | $u\_cpha(s,d)$                     | $u\_cpha(String s,double d)$                           |
| Global measurement operation   | $q\_measure()$                     | $q\_measure()$   |
| Measurement operation on the $p$ -th qubit   | $q\_measure(p)$                    | $q\_measure(int p)$                                    |
| Tensor product operation between quantum data $v_1$ and $v_2$  | $v_1 q\_tensor v_2$                | Call the member function of $v_1 q\_tensor(QType v_2)$ |
| Assignment operation from variable $b$ to variable $a$   | $a=b$                              | $a=b.transfer()$                                       |

而 QType 类的外部过程调用接口是:

```
public native void command(String cmd);
public native void command(double cmd);
public native boolean measure(int reg, int bit);
public native int measure(int reg);
```

其中 `command` 接口对应于非测量操作,参数 `cmd` 包含了整条量子汇编指令,量子汇编与解释程序汇编 `cmd` 中的指令并执行;两个 `measure` 接口分别对应于单位测量和全测量,参数 `reg` 是待测量的量子寄存器号,`bit` 是待测量的量子位.这些外部过程都在量子汇编与解释程序中实现.

为简便计,语法分析采用递归子程序方法.NDQJava 程序中量子部分仅以两种形式出现:一是类成员量子变量的说明,二是括以 `begin` 与 `end` 的量子分程序.量子变量说明有不带初始化和带初始化两种:前者处理较为简单,只需转换成类变量的说明即可;后者却需要分析初始化表达式,将之转换成 QType 中用于对其实例进行初始化的成员

函数.量子分程序的处理首先定出其 begin 与 end 匹配情况,并分析其中的量子变量说明和量子语句,细节参见文献[9,10],整个处理流程如图 2 所示.

(4) 量子汇编与解释程序的功能是将源程序中的量子部分经编译后所得到的量子汇编指令列汇编成相应之量子机器指令列,然后由量子解释程序在经典计算机上模拟解释执行.

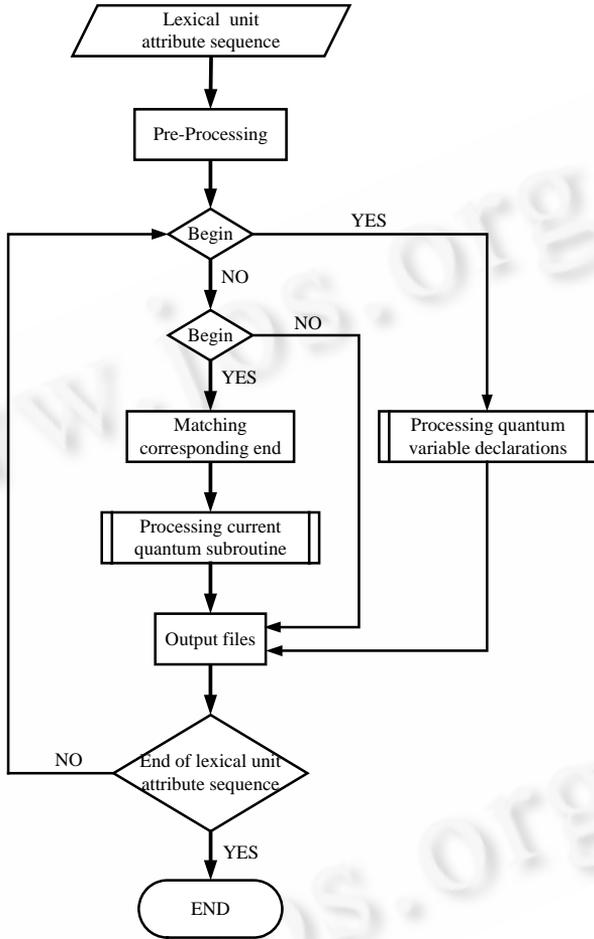


Fig.2 Flow diagram of syntactic analyzer of NDQJava processing system

图 2 语法分析子程序流程图

① 基本量子操作集

根据完备、简明、易用等原则,设置空、阿达玛、等同、X、Z、S、真条件置位、假条件置位、对换、受控非、绕 X 轴条件旋转、绕 Y 轴条件旋转、绕 Z 轴条件旋转、条件相位以及测量等 15 种基本量子操作.考虑到完备性,必须包含受控非、绕 X 轴、绕 Y 轴、绕 Z 轴条件旋转,设置其他基本量子操作则是考虑到简明、易用等原则.这些为数不多的操作在现有量子算法中出现概率颇高,其设置便于程序人员写出紧凑的程序.

② 以基本量子操作集为基础,设计了量子机器语言与量子汇编语言.二者均包含一地址指令与二地址指令.在机器语言中,一地址指令的字长为 16 位(二进制),前 4 位表示操作码,中间 4 位为量子存储的段号,最后 8 位标示所操作的量子位;二地址指令的字长为 32 位,其中后 16 位标示第 2 操作数.在汇编语言中,一地址指令使用 4 个字符,其中第 1 个字符表示操作码,其他 3 个字符标示操作数,第 2 个字符 M 标示量子存储单元号,另两个字符则标示所要进行操作的量子位.二地址指令使用 8 个字符,其中前 4 个字符的含义同前,后 4 个字符用以标

示第二操作数.

### ③ 汇编及解释

汇编程序接收量子汇编指令,将其汇编成相应的机器指令,再由解释程序对之解释执行.对酉指令,则改变量子态的概率幅;对测量指令,则回送测量结果.

## 6 示 例

Deutsch 算法.其 NDQJava 程序如下:

```
public class Deutsch
{
    public static void main (String [] args)
    {
        Deutsch d=new Deutsch();
        int i=d.deutsch_1();
        System.out.println ("The result 1 of Deutsch is:"+i);
        i=d.deutsch_2();
        System.out.println ("The result 2 of Deutsch is:"+i);
    }
    public int deutsch_1()
    {
        int i=0;
        begin
            qtype qs;
            qs:=init 2::0;
            qs:=qs u_i(1)u_x(0);
            qs:=qs u_h(1)u_h(0); //输入函数 f 为平衡函数
            qs:=qs u_cnot(1,0);
            qs:=qs u_h(1)u_i(0);
            i:=qs q_measure(1);
        end
        return i;
    }
    public int deutsch_2()
    {
        int i=0;
        begin
            qtype qs;
            qs:=init 2::0;
            qs:=qs u_i(1)u_x(0);
            qs:=qs u_h(1)u_h(0); //输入函数 f 为常函数
            qs:=qs u_i(0)u_i(1);
            qs:=qs u_h(1)u_i(0);
            i:=qs q_measure(1);
        }
    }
}
```

```

    end
    return i;
}
}

```

经过 NDQJava 处理系统处理后得到量子部分汇编指令序列如下:

```

XM00 HM01 HM00 CM01 NM00 HM01
MM00 XM02 HM03 HM02 HM03 MM02

```

由量子汇编程序汇编出如下量子机器指令并解释执行:

```

0011 0000 0000 0001    0001 0000 0000 0010    0001 0000 0000 0001
0111 0000 0000 0010    0000 0000 0000 0001    0001 0000 0000 0010
1110 0000 0000 0001    0011 0000 0000 0100    0001 0000 0000 1000
0001 0000 0000 0100    0001 0000 0000 1000    1110 0000 0000 0100

```

最终程序输出结果为

```

The result 1 of Deutsch: 1
The result 2 of Deutsch is: 0

```

与预期结果相符.

## 7 结 语

如文献[8]所述,NDQJava 语言是我们在量子程序设计领域的首次实验,其处理系统亦然,系统中仅包含了一些最为基本的内容,诸如中间语言程序之优化、基本量子操作集之选取、对应于量子数据类型 `qtype` 之 `QType` 类等均有待于进一步改进完善.

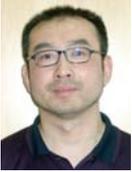
**致谢** 参加本项工作的还有刘吉、吴楠、钱辰、焦阳、徐明君、董青、朱晓瑞、颜仙乐等人,北京大学、国防科学技术大学、中创软件工程公司、苏州大学、河海大学、南京大学等单位均曾对本项目给予资助,在此一并深致谢意.

## References:

- [1] Shor PW. Algorithms for quantum computation: Discrete logarithms and factoring. In: Proc. of the 35th Annual Symp. on the Foundations of Computer Science. Santa: IEEE Computer Society Press, 1994. 124–134.
- [2] Grover LK. A fast quantum mechanical algorithm for database search. In: Proc. of the 28th Annual ACM Symp. on the Theory of Computing. ACM Press, 1996. 212–219.
- [3] Knill EH. Conventions for quantum pseudocode. LANL Report, LAUR-96-2724, 1996.
- [4] Baker GD. “Qgol”: A system for simulating quantum computations: Theory, implementation and insights [Honours Degree Thesis]. Macquarie University, 1996.
- [5] Ömer B. A procedural formalism for quantum computing [MS. Thesis]. Vienna: Technical University of Vienna, 1998.
- [6] Zuliani P. Quantum programming [Ph.D. Thesis]. Oxford: University of Oxford, 2001.
- [7] Altenkirch T, Grattage J. QML: Quantum Data and Control. 2005. <http://www.cs.nott.ac.uk/~txa/publ/jqpl.pdf>
- [8] Xu JF, Song FM, Qian SJ, Dai JA, Zhang YJ. Quantum programming language NDQJava. Journal of Software, 2008,19(1):1–8 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/19/1.htm>

## 附中文参考文献:

- [8] 徐家福,宋方敏,钱士钧,戴静安,张云洁.量子程序设计语言 NDQJava. 软件学报,2008,19(1):1–8. <http://www.jos.org.cn/1000-9825/19/1.htm>



宋方敏(1961—),男,江苏无锡人,博士,教授,博士生导师,CCF 高级会员,主要研究领域为计算机软件与理论.



张云洁(1984—),女,硕士生,主要研究领域为计算机软件.



钱士钧(1941—),男,教授,主要研究领域为计算机软件.



徐家福(1924—),男,教授,博士生导师,CCF 高级会员,主要研究领域为高级语言,新型程序设计,软件自动化.



戴静安(1980—),男,硕士,主要研究领域为计算机软件.

全国第 15 届计算机辅助设计与图形学(CAD/CG 2008)学术会议

征 文 通 知

由中国计算机学会主办、辽宁师范大学承办的全国第 15 届计算机辅助设计与图形学学术会议(CAD/CG' 2008)将于 2008 年 07 月 22 日在中国大连举行。本次会议内容包括中国计算机学会计算机辅助设计与图形学专业委员会恢复学术活动 30 周年纪念座谈、大会学术报告、计算机辅助设计与图形学热点问题专题研讨、最新成果和应用系统演示,并将邀请国内外学术界和产业界的著名专家、学者到会作特邀报告。会议录用的部分优秀论文将推荐至《计算机学报》、《计算机研究与发展》、《计算机辅助设计与图形学学报》、《工程图学学报》、《软件学报》(增刊)、《中国图象图形学报》、《系统仿真学报》发表。大会录用论文将由正式出版社出版。热诚欢迎一切从事计算机辅助设计与图形学研究、应用及软件开发的专家、学者和专业技术人员踊跃投稿。

一、会议论文主题包括(但不限于)

- |                            |               |
|----------------------------|---------------|
| 计算机辅助设计(CAD)               | 计算机辅助几何设计     |
| 几何造型与处理                    | 计算机集成制造       |
| 虚拟设计与制造                    | 网络化制造         |
| 电子设计自动化(EDA)               | 图形学基础理论与算法    |
| 科学计算可视化                    | 虚拟现实多媒体技术     |
| 计算机动画                      | 真实感图形         |
| 非真实感图形                     | 数字媒体技术与数字内容处理 |
| 图形图像融合技术人机交互技术             | 工程图形及应用       |
| 计算机图形仿真、及与计算机辅助设计与图形学的相关领域 |               |

二、重要日期

截稿日期: 2008 年 4 月 20 日

三、联系方式

投稿邮箱: cadcg2008@lnnu.edu.cn

联系人: 孙晓鹏 博士

电话/传真: 0411-82158874

相关网址: <http://www.cadcg2008.lnnu.edu.cn>