

# 基于XML的数控仿真系统的NC代码解析

牟进, 郑洪源, 丁秋林

(南京航空航天大学 计算机应用研究所, 江苏 南京 210016)

**摘要:**在直接通过NC代码来驱动仿真加工过程的数控仿真系统中, NC代码的解析处于系统的核心地位。然而各种NC代码之间的差异给系统的通用造成巨大的困难, 系统各个模块的工作严格依赖于特定的NC代码格式, 也使得系统难以扩展与变更。采用XML文档来作为用户提交的NC代码与系统之间的中间代码, 很好地解决了以上问题, 并获得了良好的应用效果, 同时, 对实现基于网络服务的数控仿真服务提供了可行性方案。

**关键词:**中间代码; XML; 文档类型定义; NC代码解析

中图分类号: TP311.52

文献标识码: A

文章编号: 1005-3751(2006)01-0102-04

## The Parser of NC Code for Simulation Based on XML

MU Jin, ZHENG Hong-yuan, DING Qiu-lin

(Institute of Computer Application, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)

**Abstract:** Parser of NC code is one of most important parts in the NC simulation system that is driven directly by NC program. However the difference among various NC code brings many problems for generalization of our system. Especially, when all modules are strictly depend on special format of NC code, it's difficult to extend or modify exist system is another point. In this paper we give a feasible solution not only to the problems mentioned above but also as a base of application for NC simulation service based on Web service. This solution is the parser of NC code for simulation based on XML that is proved effective and feasible in application.

**Key words:** medium code; XML; document type definition; parser of NC code

### 0 引言

目前国内大多数仿真系统的核心工作都是基于某种特定的NC编程规范, 如用户要采用不同的编程规范, 系统就得重复开发, 突出系统通用性差的问题。因此, 文中提出采用XML文档形式的中间代码来充当连接用户与系统的接口, 系统核心工作都是基于这种统一的XML中间代码, 有效地解决了系统通用性的问题。

### 1 数控仿真系统的体系结构分析

文中讨论的数控仿真系统由用户模块组与系统服务模块组两大部分构成。用户模块组是系统与用户的接口, 由NC程序管理模块、毛坯设计模块、刀具管理模块、CRT加工过程模拟显示模块、其它参数设置模块、系统帮助模块构成。系统服务模块组是完成系统内部数据加工处理及控制的模块组, 由NC代码解析模块、加工模拟控制模块等两个核心模块构成。整个仿真系统体系结构图如图1所示。

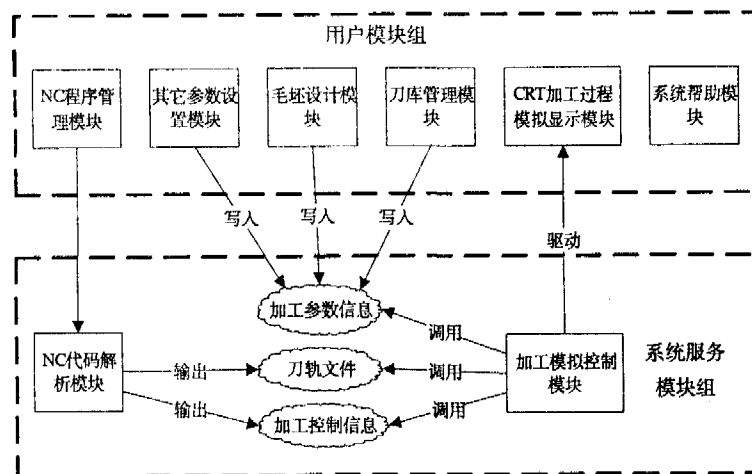


图1 数控仿真系统体系结构图

### 2 NC代码解析

在由NC代码驱动的加工运动仿真系统中, NC代码解析是系统的核心之一。NC代码解析模块通过对数控程序的分析和译码获取加工信息, 经刀具半径及长度补偿, 再经过直线插补、圆弧插补、加减速快速移动等运算得到刀具在运动中某一时刻的位置数据, 进而控制刀具图形以擦图法完成毛坯在加工过程中的改变<sup>[1]</sup>。其中NC代码解析工作原理如图2所示。

收稿日期: 2005-04-14

作者简介: 牟进(1978—), 男, 重庆万州人, 硕士研究生, 研究方向为企业信息化与系统集成。

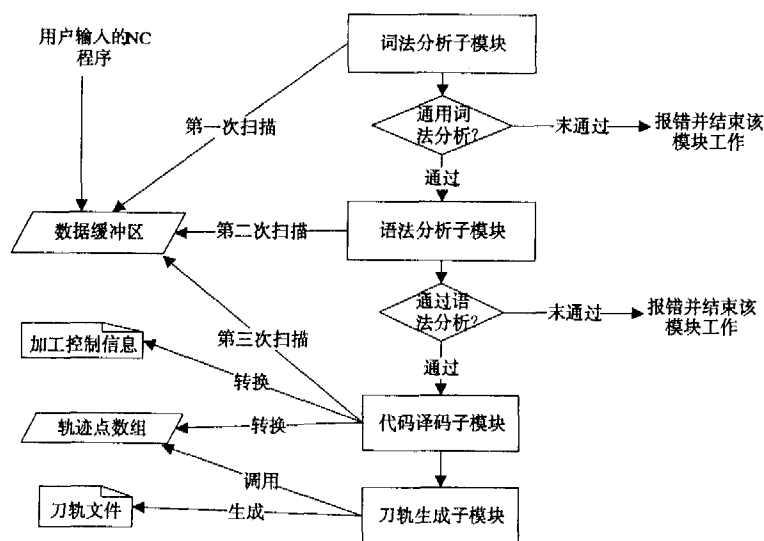


图2 传统NC代码解析工作原理图

1)词法检查子模块:判断NC代码中是否有非法字符,数值是否有效,并去除代码中多余的空格与代码中的注释行。

2)语法检查子模块:验证代码是否完整有效,比如在走刀前是否用M03主轴转动指令;程序结束后是否有M02程序结束指令;G功能字后必需的数值参数是否空缺。

3)代码译码子模块:把缓冲区里的NC代码逐行读取出来生成相应的轨迹点,存入轨迹点数组,同时把NC程序中的加工控制信息写入控制信息文件中。

4)刀轨生成子模块:根据轨迹点数组中的轨迹点信息生成刀轨文件。

代码解析完成后输出刀轨文件和加工控制文件。刀轨文件、加工控制文件再结合其它加工信息在加工模拟控制模块的控制下驱动CRT图形显示,用户最终得到的就是可视化的加工动态仿真。

### 3 基于XML的NC代码解析

#### 3.1 中间代码

通过分析,上述的NC代码解析模块仅适合某种专有仿真系统,整个解析工作过程紧密依赖于固定专有系统所采用的特定NC代码编程规范。因此,系统的扩展性并不理想,特别是系统内部所采用的信息表示通用性差。深入分析发现尽管存在不同的NC编程规范,NC程序所要表述的数据与控制信息却是如出一辙,只是表现形式上的差异而已。因此只要借用一种具有统一标准的中间代码来表述用户提交的NC程序即可解决以上问题。考虑到XML结构化、标签化、易于交换的特性,选择XML作为这种中间代码。其优点如下<sup>[2]</sup>:

1)便于描述和共享数据及文档格式,使得用XML作为中间代码成为可能。

2)便于创建、显示和维护文档。XML清晰的内部结构为更深层次的内容管理提供了可能,这种清晰的内部结构极大地有助于在文档中反应出NC程序的程序结构。

3)便于创建读写XML的应用程序。XML语法分析器是一些通用工具,可以方便地读取XML并将这些文档的内容报告给应用程序。利用这些工具可以简化系统中的语法、词法检查工作。

4)XML能创建、管理和提交数据,结合超文本传输协议(Hyper Text Transfer Protocol,HTTP)、文档对象模型(Document Object Model,DOM)与可扩展样式表语言转换(Extensible Stylesheet Language Transformation,XSLT)使最初为文档创建的工具变成了数据处理和交换的框架体系。正是这一点为实现基于网络服务的仿真系统提供了应用基础。

#### 3.2 为XML文档定义DTD

引入XML后,NC代码解析的首要工作就是将用户输入的不同NC代码转换成为XML文档,此后的工作都是基于这种统一的文档。转换工作必须按一套严格的规范进行工作,而DTD(Document Type Definition,文档类型定义)就是这套规范的基础和核心<sup>[3]</sup>。在此定义DTD就是对转换工作进行规范。通过分析,一般NC程序有如下几条基本规则<sup>[4]</sup>:

- 1)任何动作都要在程序开始后,并在程序结束前完成。
- 2)刀具的插补运动必须在主轴转的前提下完成。
- 3)主轴在开动前,工件必须夹紧。
- 4)其它动作可以在程序中任意位置出现。
- 5)NC子程序内部不能调用自身,即不能迭代。

根据以上规则给XML文档的DTD的定义如下:

```

<? xml version="1.0" standalone="no"? >
<!-- this file define the type of XML document -->
<! DOCTYPE NCcode [
<! ELEMENT NCcode( fix *, Wsubprogram *, Rsubprogram *,
locate *, method *, pause *, ncpause *, coller *, error * )>
<! ELEMENT fix( Cut *, Wsubprogram *, Rsubprogram *, lo-
cate *, method *, pause *, ncpause *, coller * )>
<! ATTLIST fix rownumber CDATA #IMPLIED>
<! ELEMENT
Cut( line *, arc *, Wsubprogram *, Rsubprogram *, locate *,
method *, pause *, ncpause *, cller * )>
<! ATTLIST Cut rownumber CDATA #IMPLIED>
<! ATTLIST Cut rotate (left|right) #REQUIRED>
<! ELEMENT Wsubprogram( fix *, locate *, method *, pause
*, ncpause *, coller * )>
<! ATTLIST Wsubprogram subname ID #REQUIRED>
<! ATTLIST Wsubprogram rownumber CDATA #IMPLIED>
<! ELEMENT Rsubprogram(subname)>
<! ATTLIST Rsubprogram rownumber CDATA #IMPLIED>
<! ATTLIST point rownumber CDATA #IMPLIED>
<! ELEMENT locate(point)>
<! ATTLIST locate path (linear|nonlinear)"linear">

```

```
<! ATTLIST locate rownumber CDATA #IMPLIED>
<! ELEMENT method(relative|absolute)>
<! ATTLIST method rownumber CDATA #IMPLIED>
<! ELEMENT pause(time)>
<! ATTLIST pause rownumber CDATA #IMPLIED>
<! ELEMENT ncpause(condition?)>
<! ATTLIST ncpause rownumber CDATA #IMPLIED>
<! ELEMENT coller(on|off)>
<! ATTLIST coller rownumber CDATA #IMPLIED>
<! ELEMENT rotate(left|right)>
<! ATTLIST rotate rownumber CDATA #IMPLIED>
<! ELEMENT line(point,speed?)>
<! ATTLIST line rownumber CDATA #IMPLIED>
<! ELEMENT arc(point,radius,rotate,speed?)>
<! ATTLIST arc rownumber CDATA #IMPLIED>
<! ELEMENT point(xpoint,ypoint,zpoint)>
<! ATTLIST point rownumber CDATA #IMPLIED>
<! ELEMENT error(#PCDATA)>
<! ATTLIST error rownumber CDATA #IMPLIED>
<! ELEMENT xpoint(#PCDATA)>
<! ELEMENT ypoint(#PCDATA)>
<! ELEMENT zpoint(#PCDATA)>
>
```

3.3 基于 XML 的 NC 代码解析

引入中间代码后,用户选择的 NC 程序变更时,只需要重新构建 XML 文档转换子模块,其它模块均无需改动,更可以在系统内集成多种 XML 文档转换子模块来适应不同用户的需求,从而提高了系统的通用性。引入 XML 技术改进后的 NC 代码解析模块结构如图 3 所示。

3.3.1 XML 文档转换子模块

理论上说,转换过程中不必关心 NC 程序的正确性,只需按一定的规则完成文档转换。而在该系统中,为了给用户准确详细的错误信息,一部分的词法检查是在转化过程中同时完成的。

转化机制本身并不复杂,借助 Visual C++ 所提供的 CRichEdit 控件把缓冲区中的字符逐行读入—分析—转换—写入。文中以南京某数控有限公司的 NC 程序为例,依据先前定义的 DTD,转换规则如下:

- 1)当读入行号时,如 N001,记录当前行号并暂存在一个临时变量 CurrentRow 中。
- 2)当扫描一行时遇到第一个非法字符时,在 XML 文档中加一行 < error rownumber = "currentrow" > <![CDATA[error message,error row,error type]]> </error

>。其中 currentrow 是临时变量 CurrentRow 中保存的行号。

3)当读入有效功能字时,以 G00 X24.3 Y25.5(快速定位)为例:读到 G00 时在缓冲区内写入 <locate rownumber="current row"></locate>。如在 G00 后读到 X,Y,Z 三个字符的任意之一时,在缓冲区中写入 <point>x,y,z</point>,其中 x,y,z 分别是 X,Y,Z 的值,缺省时写为 NULL。因此以上代码经转换并添加入 XML 文档中的形式为 <locate rownumber="current row"><point>24.3,25.5,NULL</point></locate>。其它功能字类似。

4)当读入无效功能字时,如 G44,系统暂无此功能,在 XML 文档中加一行 <error rownumber="currentrow"> <![CDATA[error message,error row,error type]]> </error>。

5)当读入辅助功能字时,如 M03(主轴正转),此时写入一行 <cut rotate="left">,只有在执行 M20(主轴停)时再写入一行 </cut>。

以上罗列出了文档形式转换的几条基本规则,在编写调试程序时还有许多需要注意的小细节,这里就不再详述。

在文中介绍的系统中预定义了 FANUC、SIEMENS 两个常用控制系统的转换子模块供用户选用,需要时还可为系统扩展新的子模块,用户可在参数设置模块中选择。图 4 所示为用户选择窗口。

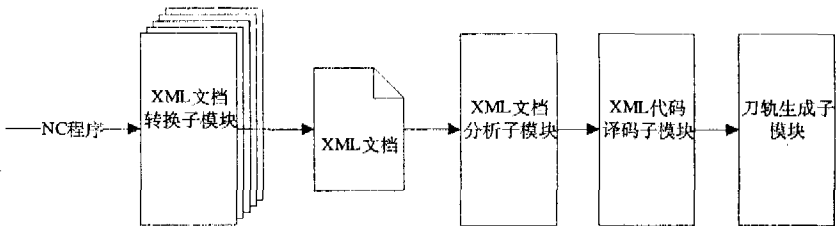


图 3 引入 XML 后的 NC 代码解析模块结构

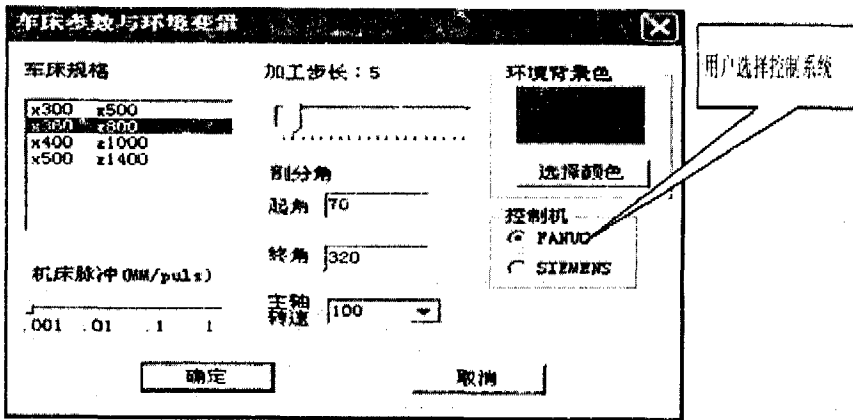


图 4 车床参数与环境变量设置窗口

3.3.2 XML 文档分析器子模块

引入 XML 作为中间文档后,可以选择通用的 XML 语法分析器加以改造,重新定义应用接口就可得到所需的 XML 语法分析器。这样避免了大量重复的开发工作。在前面的 DTD 中已经定义了 XML 数据结构,

此时的语法分析工作就是检查中间文档的结构是否良好,文档是否符合在文档类型定义(DTD)中的规范。

再以南京某数控有限公司的 NC 程序为例:当 NC 程序中缺少 M03(主轴正转)时,又或是 M03 与 M20(主轴停)次序颠倒时,XML 文档中一定会出现 `</cut>` 或 `</cut><cut>` 这样的结构不完整或结构错误。这种结构上的错误可以完全依赖 XML 语法分析器检查出来。

### 3.3.3 基于 XML 文档的代码译码子模块

上述语法分析器实为一个校验语法分析器,通过分析文档一定是有效的、有着严格数据结构的 XML 文档。代码译码工作就是把 XML 数据中各种信息分离出来,转化为轨迹点存储到轨迹点数组中。基于 XML 文档的代码译码工作机制与传统的代码译码相同,二者的区别仅在于前者所采用的源文件已经是统一的 XML 文档,后者则是用户直接提交的 NC 程序。其后的刀轨生成子模块也可完全沿用传统 NC 代码解析中的相应子模块。

文中分别采用 FANUC 与 SIEMENS 两种不同格式的 NC 代码对同一零件进行加工仿真,通过验证得到了预期的效果,两种情况下模拟加工的过程和结果完全相同,如图 5 所示。

## 4 结束语

文中着重介绍了 XML 技术在 NC 代码解析中的应用,其关键又在如何将用户提交的 NC 程序转换成为 XML 文档。而将 XML 技术引入到数控仿真系统中的意义还在于它为实现基于网络服务的数控仿真服务系统提供了

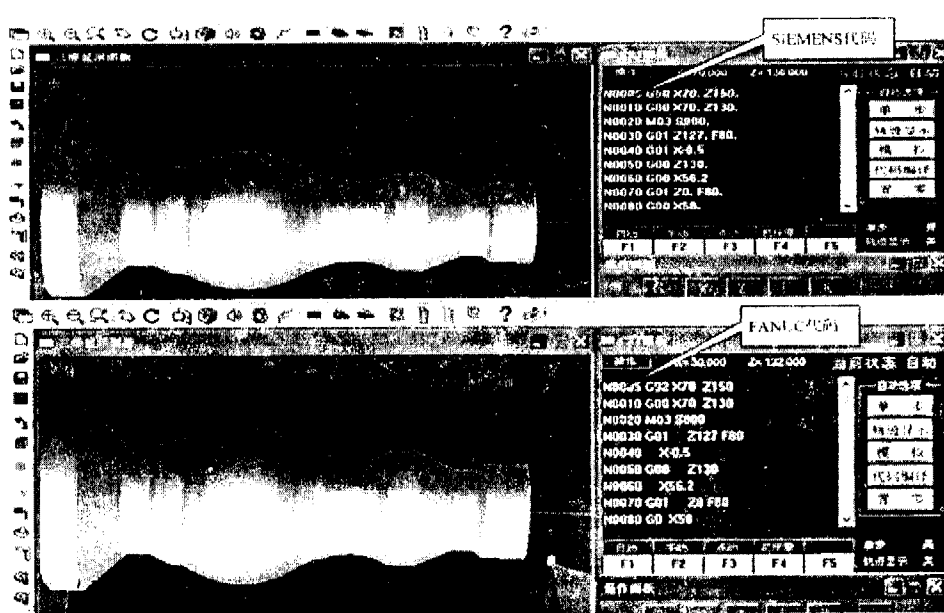


图5 FANUC代码与SIEMENS代码模拟加工结果对比

可行的方案。在文中所述应用的基础之上,再将系统其它模块的输入输出信息均采用 XML 文档格式表示,为实现网络仿真服务系统提供了应用基础,很好地解决了通过集成本地系统服务来构建仿真系统带来的系统和外部交互性差、开发周期长、重构性差等问题<sup>[5]</sup>。

### 参考文献:

- [1] 李宏胜. 数控原理与系统[M]. 北京:机械工业出版社, 1998.
- [2] Goldfarb C F, Prescod P. XML 实用技术[M]. 张利译. 北京:清华大学出版社, 1999.
- [3] Harold E R, Means W S. XML 技术手册[M]. 孔小玲译. 北京:中国电力出版社, 2001.
- [4] 方沂. 数控机床编程与操作[M]. 北京:国防工业出版社, 1999.
- [5] Zoo J, Chen Y P, Zhou Z D. Building Open CNC Systems with Software IC Chips Based on Software Reuse[M]. [s. l.]: Springer Press, 2000.

(上接第101页)

## 6 小结

文中就车牌侧视的几何变形问题进行了探讨,利用双线性映射将变形的车牌图像还原为正视的矩形车牌图像,采用了全选主元高斯消元法来求该双线性变换方程组的解,准确而快速地解决了车牌图像的变形矫正问题。

### 参考文献:

- [1] 陈锻生, 谢志鹏. 复杂背景下彩色图像车牌提取与字符分割技术[J]. 小型微型计算机系统, 2002, 23(9): 1144 -

1146.

- [2] 屈磊, 王年, 吴冠勇, 等. 基于色彩饱和度和字符特征的汽车牌照定位[J]. 微机发展, 2003, 13(9): 6 - 7.
- [3] 段震, 姚芳兵, 张铃. 基于构造性学习方法的车牌定位[J]. 微机发展, 2004, 14(8): 41 - 42.
- [4] 曲新亮, 魏守水, 杜洋. 一种基于 Gabor 小波变换的图像倾斜矫正算法[J]. 微机发展, 2004, 14(11): 19 - 20.
- [5] 阮秋琦. 数字图像处理学[M]. 北京:电子工业出版社, 2001. 330 - 332.
- [6] 廖晓钟, 赖汝. 科学与工程计算[M]. 北京:国防工业出版社, 2003. 69 - 88.