

The Realization of the 3D Cloud Printer Operation Based on the Internet

Xiuli Shao¹, Guanghui Chen², Caixia Liu¹, Jin Niu¹

¹College of Computer and Control Engineering, Nankai University, Tianjin

²Rapid Prototyping Center, Tianjin University, Tianjin

Email: shaoxl@nankai.edu.cn, chen_guanghui@tju.edu.cn, katrina_2010@126.com,
james_niu@mail.nankai.edu.cn

Received: Mar. 10th, 2015; accepted: Mar. 24th, 2015; published: Mar. 30th, 2015

Copyright © 2015 by authors and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

UP is a stand-alone software, which is used to edit 3D models and drive 3D printer for printing. It first captures the edit command of model files which are built in the machine and its control command to the driver of printer. Then the commands should be encapsulated for transmitting on the Internet, to support UP operation and drive 3D print based on Internet. Thus user can use 3D print resources scattering at different places.

Keywords

Cloud Print, UP, Command Parsing, Capture Model Editing Commands, Capture Print Command

基于Internet操作3D云打印机的实现

邵秀丽¹, 陈光辉², 刘彩霞¹, 牛 津¹

¹南开大学计算机与控制工程学院, 天津

²天津大学快速成型中心, 天津

Email: shaoxl@nankai.edu.cn, chen_guanghui@tju.edu.cn, katrina_2010@126.com,
james_niu@mail.nankai.edu.cn

收稿日期: 2015年3月10日; 录用日期: 2015年3月24日; 发布日期: 2015年3月30日

摘要

对3D设计模型进行编辑和驱动3D打印机进行打印的单机软件，例如UP，先捕获其对安装在机器中的模型文件的编辑命令，以及其对打印机的驱动控制命令，然后对这些命令分别解析，并封装为Web化，以支持基于Internet实现对UP操作和对3D打印机进行驱动，实现了用户基于互联网使用散落在不同地方的3D打印机资源。

关键词

云打印，UP，命令解析，捕获模型编辑命令，捕获打印命令

1. 概述

论文平台充分利用互联网优势，也充分利用闲散的自主3D打印机资源，为广大3D设计爱好者、设计提供方、3D打印资源拥有者、系统管理方提供一个交流与协作的服务平台，让他们在新媒体的大环境中，能够很快的把自己的知识转化成生产力，并进一步促进产品设计的快速成型[1]。3D打印机在工业、医疗、建筑等行业都有高可用性，因此对3D云打印的研究是十分必要和有意义的。

3D云打印服务平台的设计与实现已有另一篇文章详细介绍，本文介绍打印过程中，针对3D打印机任务要求的实时性、安全性与高效性，从如何捕获对其机器上模型文件的编辑命令和驱动打印机的打印命令，并将命令封装Web化和解析执行这几个方面研究了3D打印机接入平台的实现方法。编辑命令和驱动打印机命令的捕获、封装与解析的实现用相同的协议进行传输，但是执行两个命令的驱动对象不同。编辑命令经解析驱动的是UP软件，由UP软件执行命令；打印命令经解析后，是由UP软件执行驱动3D打印机。

系统故障或者是网络故障都会对3D云打印操作产生很大的影响，本论文就命令传输过程中可能出现的故障提出了解决方案，以确保当故障发生时，不会造成特大损失或给用户带来恶劣体验。

2. 操作3D云打印机命令的传输

多个用户通过Internet访问云端平台提供的3D打印服务，向Web服务器发送请求，3D打印机接入Internet平台，主要依靠操作命令在网络上的流转实现。以用户发送打印机驱动命令为例，打印机驱动命令在Internet上的流转如图1所示。

图中实线箭头表示驱动命令在公网上的流转方向，虚线表示打印机返回给用户信息的流转方向。

Step 01 用户通过Web浏览器向服务器发送打印机驱动命令，按照HTTP协议传输，将打印机驱动命令封装成易于传输的指令数据；

Step 02 Web服务器接收驱动命令，并将指令数据集存储在临时资源池中，利用Web服务器缓存机制，可以存储命令；

Step 03 经测试知道服务器端的临时资源池并不能持续安全的存储数据，有可能出现打印命令还没有执行的情况下命令丢失，这样会造成用户对模型操作命令丢失情况，为了避免这种情况的发生我们利用异步存储的机制将用户对模型的编辑指令数据集存储在数据库服务器的数据表中，防止临时资源池中的数据丢失；

Step 04 待工作站上的守护进程，捕获到Web服务器端的驱动命令，Web服务器通过HTTP协议将指令数据集发送给工作站。

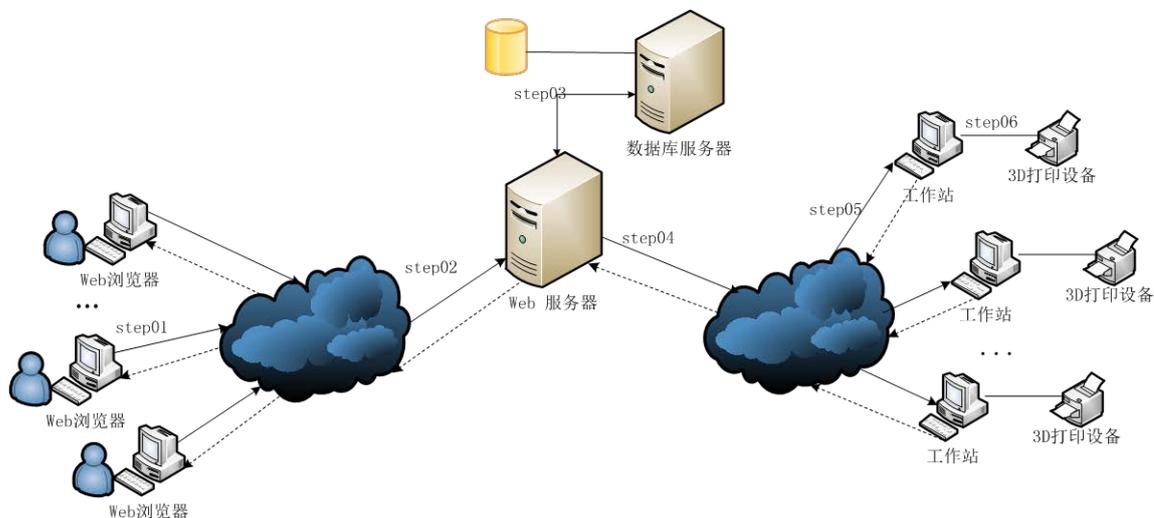


Figure 1. 3D cloud printer driver command flow chart on the Internet
图 1. 3D 云打印机驱动命令在 Internet 上的流转图

这里不采用服务器向工作站发送请求的方式，而是工作站上的守护进程[2]不断请求服务器，是因为与 3D 打印机直接相连接的工作站很可能没有可供外网访问的对外 IP 地址，更可能由于 NAT 和两端主机所在的防火墙等原因而无法从 Web 服务器向客户端直接建立 TCP 连接来传输数据[3]，因此，所设计的协议应当采用由客户端向 Web 服务器建立连接的方式来进行通信，此时，Web 服务器同时充当了 TCP 服务器，由于 HTTP 协议也以 TCP 协议为传输层协议，Web 服务所用的 80 端口在服务器上也必然打开；因而，一种可行的替代方案(也是最终实现时所采用的方案)是，以 HTTP 请求来代替建立 TCP 连接，以 HTTP 为基础协议来传输需要封装的字节流数据；

Step 05 工作站接收 Web 服务器发送的指令数据集，按照 HTTP 协议格式解析指令(HTTP 协议格式在第 3 章详细介绍)，按照控制指令模拟人员操作 UP 软件，执行相应的操作；

Step 06 操作命令若为打印命令，则由 UP 软件调用与打印机的接口，执行打印命令。

按照虚线箭头的方向，工作站端通过获取 UP 软件底部状态栏信息，将打印信息按照协议封装，经过网络传输发送给 Web 服务器，服务器将打印信息保存在数据库服务器上的数据表中，待用户查询打印状态时，将任务打印状态信息显示在 Web 浏览器上，供用户查看。

若用户执行的是编辑命令，则通过将 UP 软件界面置顶，然后截屏的方式，将执行结果以图片的形式，传输至 Web 浏览器端呈现给用户。

3. 3D 打印机接入 Internet 平台的实现

由 3D 打印机驱动命令在 Internet 上的流转图可知，打印命令应该以一种安全且高速的方式在网络平台上传输。为了保证客户端与服务器的快速交互，我们采用 HTTP 协议传输操作指令，同时我们还需要考虑数据的安全性。我们采用虚拟工作站协议(Virtual Workstation Protocol, VWP)传输操作指令，用 HTTP 协议代替 TCP 协议作为传输协议，既保证了指令传输的高效性同时保证了数据的安全性。由于服务器端的内存保护机制，操作命令在服务器临时资源池中并不能存储很长时间，为了防止操作命令丢失，采用数据库服务器的备份方式保证用户操作不会丢失。

3.1. 操作 3D 打印机命令的 Web 化

为了快速传输 3D 打印机的驱动命令，我们采用 HTTP 协议封装的打印命令。同时考虑到 3D 打印编

辑操作的有限性和操作参数的有限性，因此，命令的传输以 HTTP 协议进行传输的，HTTP 消息有客户端到服务器的请求和服务器到客户端的响应组成。对打印机的控制也是由客户端发起，建立连接，等待服务器响应，返回客户端 Web 页面信息。由于 HTTP 协议支持客户/服务器模式，客户向服务器请求服务时，只需传送请求方法和路径。命令的传输采用 post 方法，可以传输任意类型的数据对象，正在传输的类型由 Content-Type 加以标记。HTTP 协议的传输比较灵活，只需要客户端与服务器两端以统一的格式进行封装与解析即可[4]。

对于操作命令的封装需要按照一定的格式，HTTP 协议的格式如下：

Format	Type	Parameters
1 字节	1 字节	多字节

HTTP 协议封装之后，传输数据正文格式为：format = X & type = X & parameters = X..。format 表示传输数据类型，type 表示执行操作类型，parameters 表示参数值。Format 和 type 可以用短整型传输参数，不同的数字代表不同的操作类型。以用户发送一次将模型扩大 3 倍的命令为例，由于缩放存在参数类型为负数的情况，且缩放比例有上限和下限要求，所以参数类型应该为有符号短整型数据，即取 format = 3；缩放视图对应的操作号为 5，因此 type = 5；放大 3 倍，则 parameters = +3。封装之后传输数据格式应该为：format = 3 & type = 5 & parameters = +3。

由于不同的用户会发送不同的命令，服务器会同时接受到很多命令，为了区分用户发送的命令，我们可以再传送一个 uid 参数，将用户 id 作为唯一标识符区分不同用户的发送的命令。

3.2. 操作 3D 云打印机命令的解析

每次工作站客户端向 Web 服务器进行请求之后，收到的响应内容都是如上格式定义的。3D 打印机在执行操作命令之前必须在工作站端对操作指令解析。工作站客户端程序请求 Web 服务器，有 HttpWebResponse 的对象 GetResponseStream()方法获取 Web 服务器响应的数据信息流，初始化 StreamReader 对象，使用 StreamReader 对象的 ReadLine()方法获取服务器端响应命令数据，从而完成命令数据的接收。

工作站端接收到操作命令之后，需要按照协议格式进行解析[5]，Encoding.UTF8.GetBytes()方法获取传输数据正文，解析得到 format，type，parameters 对应的参数值。此时，获取的数据类型不是 string 类型，需使用 toString()方法将参数转换为 string 类型，然后用 switch 语句匹配数值，执行相应的操作命令。首先获取 format 的值，匹配得到参数值类型，初始化存放参数值的形式参数。其次获取 type 的值，得到操作类型。最后获取参数值。模拟人员操作 UP 软件。以传输模型放大三倍的操作命令为例，操作命令在网络间以二进制流的形式传输，工作站端接收以后，按照固定的 HTTP 协议格式解析，分别得出 format 值为 3，即传输数据类型为有符号短整型数据；type 值为 5，即操作方式是缩放模型；parameters 的值为 +3，即缩放倍数为+3，即为将模型放大 3 倍。

操作命令经过解析之后，获取到用户要执行的操作，然后向 3D 打印机控制软件(UP 软件)发送控制命令，执行模型编辑操作。如何模拟人员操作 UP 软件，是本平台要解决的另一个难点。工作站上的 UP 软件并不存在实际的键盘，我们采用最直接的模拟方法就是：直接伪造一个键盘消息发给目标程序[6]。Windows 提供了可以直接向目标程序发送消息的 API 函数，SendMessage()函数。

具体实现方式是，先用 FindWindowEx 这类 API 函数找到 UP 程序的句柄，再向它发送键盘消息。将操作命令发送到 UP 软件对应的控件，UP 软件执行相应的操作。例如 UP 软件接收到将模型放大三倍

的命令,则将数值+3 赋值给 UP 软件对应的缩放控件,模拟人员执行操作。UP 执行完模型放大操作之后,将执行结果以图片的形式经过 HTTP 协议封装,发送给 Web 服务器,服务器再返回给 Web 浏览器。

3.3. 操作 3D 打印机命令的存储

3D 打印模型会用于工业、医疗等行业,对模型打印的精准性有很高的要求,因此,必须准确安全的保存用户的操作和选择。但是由于服务器端的内存保护机制,操作命令在服务器临时资源池中并不能存储很长时间,为了防止操作命令丢失,采用数据库服务器的备份进行恢复最初的状态。

3.3.1. 请求命令的存储

操作命令交互表:

userId	userIP	workIP	dataList	new
1	192.168.0.2	201.102.0.30	****	0
2	192.168.0.100	201.102.20.30	****	1

userId 存储发送命令用户的 Id, userIP 存储用户发送该命令的当前 IP 地址, workIP 存储接收操作命令的工作站的 IP, dataList 存储操作命令, new 字段存储当前命令是否执行。

用户向服务器发出操作 3D 打印机命令请求,服务器接收后会在命令交互表中添加新的数据,包括发送请求用户的 Id,发送请求用户的当前 IP 地址,操作命令,是否执行(初始为 1,表示未执行)等信息。服务器检查命令交互表相应条目的 new 值,判断当前的命令是否已发送至工作站,然后修改 new 字段的值为 0(已执行)。Web 服务器端通过判断 workIP 值是否为空,确定是否已分配编辑模型的权限。获取权限后,用户发送各种编辑命令,保存到命令交互表中相应 userId 对应的 dataList 字段中,同时修改 new 字段的值为 1(未执行),并把数据存储到数据库相应表中,以防止数据的丢失。

3.3.2. 打印状态的存储

状态交互表:

userId	userIP	workIP	dataList	new
1	192.168.0.5	201.102.0.10	****	1

userId 存储发送打印命令的用户 id, userIP 存储用户发送打印命令时的 IP 地址, workIP 存储执行打印命令的工作站的 IP 地址, dataList 存储打印状态, new 表示当前打印状态是否被用户查看过。

工作站端会把打印任务当前的状态,以合理的间隔发送到服务器端,服务器端通过检索状态交互表,修改相应条目的 dataList 值,同时修改 new 值为 1(未取),然后把数据保存到数据库。

当用户通过 Web 浏览器发送状态查看请求时,服务器根据用户 userId 字段检索状态交互表,将 new 字段值为 1 的 dataList 数据发送至浏览器端。

4. 操作 3D 云打印机命令故障处理

为了保证用户能够正常通过 Internet 操作 3D 打印机,我们做了必要的故障预处理工作。操作打印机命令发出后有可能出现异常,客户端与服务器之间网络中断、服务器故障、工作站故障等原因导致控制命令无法被执行。为了保证用户打印命令发出后能够正常被执行,我们做了以下工作:

客户端无法与服务器连接的情况下,用户的编辑命令或者打印命令无法被服务器接收,这种情况下

的解决办法只有等待网络恢复之后，再执行编辑操作或者打印操作；

用户打印命令发出后，打印命令会存放在服务器上的临时资源池，同时利用异步存储[7]的方式存放在数据库服务器的上的操作命令交互表中，当服务器故障导致静态数据丢失时，可以通过数据库恢复数据；

由于用户打印过程中，打印进度同时是存储在 Web 服务器端的打印机状态交互表中，所以在工作站故障的情况下，打印状态保存在数据库服务器上，待工作站恢复工作之后，可以从数据库中找到打印状态，继续执行打印操作。

5. 结论

3D 服务平台的打印命令开始由用户通过 Web 浏览器发送，将用户操作封装成为一种易于网络传输的格式，经过 Web 服务器传送至工作站，工作站接收到操作命令之后按照协议格式进行解析，然后模拟人员操作 UP 软件，执行操作命令。工作站利用 UP 软件连接打印设备，执行打印命令。Web 浏览器与 Web 服务器之间通过网络进行命令的传送；工作站与打印设备之间命令的流转依靠 UP 软件作为打印机的驱程；而 Web 服务器与工作站之间命令的流转，则依靠在工作站的守护进程。工作站的实现保证了整个 3D 服务平台命令流转的畅通。

基金项目

天津市自然科学基金资助项目(No. 12ZCDZGX46500)、(No. 13ZCZDGX01108)、(13ZCZDGX02500)。

参考文献 (References)

- [1] 吴丽, 余文春 (2015) 基于多服务器最优配置的云计算利润最大化技术研究. *计算机应用研究*, **1**, 194-197.
- [2] 王洋涛, 唐敏, 王媛媛 (2014) Windows 应用程序守护进程的实现. *电脑编程技巧与维护*, **21**.
- [3] 周志勇 (2014) 云计算环境下网络信息安全技术发展研究. *中国高新技术企业*, **25**, 40-41.
- [4] 杨国海 (2014) 基于多层协议理解的 HTTP 数据还原技术研究. *电子世界*, **22**.
- [5] Parsiegl, V., Rein, S., Wolisz, A., et al. (2011) A network aware protocol for switching HTTP streams between devices. *IEEE International Conference on Consumer Electronics (ICCE-Berlin)*, Berlin, 2011, 74-78.
- [6] 潘爱民 (2014) Windows 内核原理与实现. 电子工业出版社, 北京.
- [7] Wu, X., Turner, D., Chen, C., et al. (2012) NetPilot: Automating datacenter network failure mitigation. *ACM Sigcomm Computer Communication Review*, **42**, 419-430.