International Journal of Multimedia and Ubiquitous Engineering
Vol.10, No.5 (2015), pp.77-90
http://dx.doi.org/10.14257/ijmue.2015.10.5.09

Combing Customizable Configuration and Interactive Control to Simulate Juicing and Filling Production Line

Changyu Liu [1], Dapeng Li [2, *], Bin Lu [3], Tiezhu Zhao [4]

[1] *School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China*
[2] *College of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China*
[3] *School of Computer Science, Wuyi University, Jiangmen 529020, China*
[4] *Computer College, Dongguan University of Technology, Dongguan 523808, China*
yezhich@gmail.com, [*]dapengli@njupt.edu.cn (corresponding author), lbscut@gmail.com, tzzhao83@163.com

Abstract

Both the rapid growth of global economy and the rapid improvement in living conditions have witnessed an explosive requirement of foods that are rich of vitamin. Juice beverage, as one kind of these foods, is very popular among people of all ages due to its higher portability and easier preservability. However, there are many issues, such as a very low device utilization rate, a very long design period and an inflexible manufacturing process, by using the traditional juicing and filling production line. The introduction of virtual reality simulation technologies can not only detect bottlenecks of the production line during design period, but also process fruits according to seasons for improving the total efficiency of one single production line. In this paper, we combine customizable configuration and interactive control to design and implement a simulation system of juicing and filling production line based on the traditional client-server architecture. Practical applications show that the developed simulation system can not only shorten effectively the design period but also improve greatly the design quality.

Keywords: *customizable configuration, interactive control, juicing and filling production line, design period, client-server architecture*

1. Introduction

There are many shortages existed in current mainstream production line, such as high manufacturing cost, high maintenance cost, long design period, low facility utilization, and non-modular. Due to limited funds, various fruit categories and fruit seasonality, most of small enterprises are unable to use the mainstream production line, requiring a small sized juicing and filling production line which could adapts modularly to customer requirements.

In this paper, we try to solve above shortages by proposing a new virtual reality based multifunctional and modular design approach, which could take into account specific requirements of customers, and combine customizable configuration as well as interactive control to design and implement a simulation system of juicing and filling production line based on the traditional client-server architecture.

Comparing to traditional design approaches, the proposed one could improve the success rate of the primary development and decrease the development cost by adopting simulation technologies and manufacturing data to model and analyze the design of the production line.

2. System Analysis and Design

2.1. Requirements Analysis

Based on the client-server architecture, the developed simulation system of juicing and filling production line should provide functions of application server, FTP server, database management and order management in the server and functions of customizable configurations and interactive controls by flexible and required simulations in the client. Besides, our simulation system should meet requirements of:

(1) Parameterized customer requirements. From previous experience, most customers express their demands of configuration design for the production line either by spoken language or by other unprofessional ways. Thus, many discussions among customers and designers should be proceeded before final solutions, causing many restrictions. By introducing parameterized customer requirements, we can shorten the period of configuration design, measure required devices accurately and deal with the feedback information of customers in time.

(2) Intuitive devices selection. In order to show intuitively device models and device parameters to our customers, we are required to guide customers to configure modularly the production line by means of knowledge reuse and to realize dynamic simulations of production line models.

(3) Intelligent quotation system. During a traditional quotation process, estimators could assess the value of the production line, design layouts in detail and conduct a great deal of computation when they obtain technical drawings and are familiar with customer requirements, according to many attributes of production line device, such as appearance, structure, function, size and complexity. Traditional quotation approaches rely highly on experiences of estimators and are thus not suitable to novices. Based on the EON control, the proposed quotation system could provide efficient and intelligent interface to customers.

(4) Modularized simulation system. We adopt modularized ideas to design key devices and virtual reality technologies to display intuitively key devices to customers. The final simulation system could meet well module restructuring requirements of our customers.

2.2. Database Design

As the role of requirements analysis, database design is also an important process. In this project, we use the Oracle database and construct 8 tables.

(1) Customer requirements table: SZSCX_SER_CusRequirement, which is used to record requirement details of the production line from our customers.

Table 1. Customer Requirements Table

Field Name	Data Type	Description
User_Name	Text	User name, such as Changyu Liu
Config_Name	Text	Configuration name, such as pro_01
Fruit_Type	Number	Fruit type, such as apple
Container_Type	Text	Container type, such as glass
Container_Volume	Text	Container volume, such as 22 ml
Guanzhuang_Speed	Number	Filling speed, such as 10 per minute
Guanzhuang_Error	Number	Filling error, such as 0.33 ml
PreRequire_Price	Number	Expected price, such as $ 100, 000
……	……	……

(2) Configuration results table: SZSCX_SER_ConfigResult, which is used to record configuration details of our customers.

Table 2. Configuration Results Table

Field Name	Data Type	Description
User_Name	Text	User name, such as Bin Lu
Config_Name	Text	Configuration name, such as pro_02
Device_Type	Text	Device type， such as filling machine
Device_ModelNum	Text	Device model number, such as XM-521
……	……	……

(3) Device parameters table: SZSCX_SER_DeviceParam, which is used to record parameters of devices in the production line, such as device types, production rate, device price and device error.

Table 3. Device Parameters Table

Field Name	Data Type	Description
Device_Type	Text	Device type， such as pitting machine
Device_ModelNum	Text	Device model number, such as XM-570
Device_Speed	Number	Production rate, such as 10^6 ml/h
Device_Error	Number	Device error, such as 0.33 ml
Device_Price	Number	Device price, such as $ 131, 400
……	……	……

(4) Production line layout table: SZSCX_SER_ProLineLayout, which is used to record information about the layout of a configured production line that is determined by the fruit types.

Table 4. Production Line Layout Table

Field Name	Data Type	Description
Fruit_ID	Number	Fruit ID, such as 1
Tag_LineType	Number	Where: 0-Juicing line, 1-Filling line
Device_Type	Text	Device type， such as conveying belt
……	……	……

(5) Fruit types table: SZSCX_FruitType.

Table 5. Fruit Types Table

Field Name	Data Type	Description
Fruit_ID	Number	Fruit ID, such as 2
Fruit_Type	Text	Fruit type, such as apple for ID 2
……	……	……

(6) User information table: SZSCX_SER_USERINFO, which is used to record the registered users of our system.

Table 6. User Information Table

Field Name	Data Type	Description
User_Name	Text	User name, such as Huiling Li
Password	Text	User password, such as password_1
RoleID	Text	User role, such as administrators
……	……	……

(7) Main order table: SZSCX_SER_Main_Order, which is used to record information of the receiver and the current status.

Table 7. Main Order Table

Field Name	Data Type	Description
Order_Number	Text	Order ID, such as proline_2014_12345
Receiver_Name	Text	Receiver name, such as Hongjun Wang
Ship_Address	Text	Shipping address: such as 20 Virginia Court
Pay_Method	Text	Payment method, such as online
Ship_Method	Text	Shipping method, such as FedEx express
Total_Price	Number	Total price, such as $ 432,100
User_Name	Text	User name, such as user_1
Order_Status	Number	where: 0-processing, 1-paid, 2-cancelled
……	……	……

(8) Detailed order table: SZSCX_SER_Detail_Order, which is used to record details of selected commodities for each order.

Table 8. Detailed Order Table

Field Name	Data Type	Description
Order_Number	Text	Order ID, such as proline_2014_54321
Device_Type	Text	Device type，such as cracking machine
Device_ModelNum	Text	Device model number, such as XM-123
Device_Price	Number	Device price, such as $ 512, 000
Device_Num	Number	Device number, such as 10
……	……	……

3. Server Implementation

3.1. Application Server

Based on multi-tier distributed application services suite (MIDAS), C++ Builder could provide a mechanism for exchanging database information among client applications and server applications. For MIDAS based applications, the dynamic linking library MIDAS.DLL is used to manage the data of both clients and servers.

There are three protocols for MIDAS based communications among clients and servers, which are DCOM, Windows sockets (TCP/IP) and HTTP. The corresponding C++ Builder controls for these three protocols are TDCOMConnection, TSocketConnection and TWebConnection. In this paper, we use TSocketConnection control for communications. Figure 1 shows TsocketConnection based communication principle.

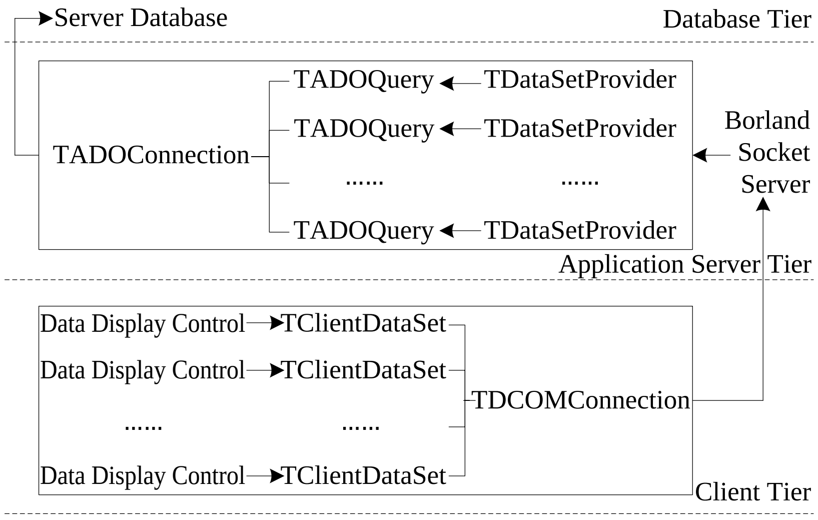
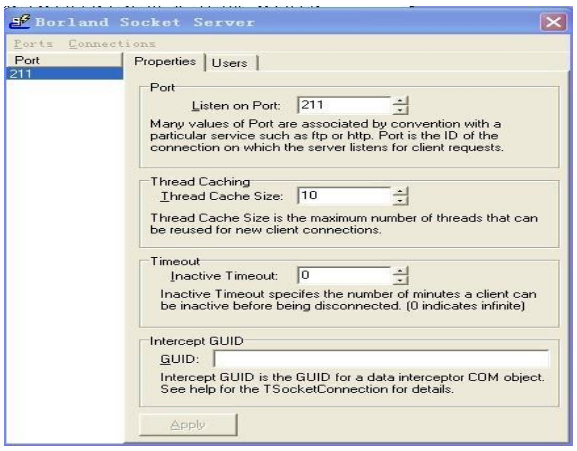


Figure 1. Communication Principle of TsocketConnection based Applications by Using MIDAS Framework

The Borland Socket Server, as shown in Figure 2, is a C++ Builder control that is developed by the Borland Corporation for TSocketConnection based applications.



Port: 211

Port
Listen on Port: 211
Many values of Port are associated by convention with a particular service such as ftp or http. Port is the ID of the connection on which the server listens for client requests.

Thread Caching
Thread Cache Size: 10
Thread Cache Size is the maximum number of threads that can be reused for new client connections.

Timeout
Inactive Timeout: 0
Inactive Timeout specifes the number of minutes a client can be inactive before being disconnected. (0 indicates infinite)

Intercept GUID
GUID:
Intercept GUID is the GUID for a data interceptor COM object. See help for the TSocketConnection for details.

Figure 2. The Database Service Agent Application for Borland Socket Server

Before connecting the server database, the client needs to call shell command to initialize scktsrvr.exe, which is a database service agent application for Borland Socket Server. Then, we can create a new application server by following steps:

(1) Firstly, create and save a new project in C++ Builder. Secondly, click File|New on the main menu to open New Items dialog box. Thirdly, select the Remote Data Module on the Multitier page of the New Items dialog box, if we want to adopt above three protocols, which are DCOM, HTTP and Sockets, as our MIDAS based applications.

(2) Add appropriate dataset controls to the Remote Data Module and set their attributes for accessing the database server.

(3) For each dataset control in the Remote Data Module, add a TDataSetProvider control, which is used to process client requests and encapsulate communication data.

(4) For each TDataSetProvider control, set their attributes, such as the Dataset attribute.

(5) Write application server codes to achieve required functions, such as event handling, service logic handling, data integrity check and data security check. Then, realize some other necessary interfaces of the application server.

(6) Save, compile and register or install the application server.

(7) If the DCOM protocol is not adopted in the application server, additional software must be installed to act the role of Remote Data Module for processing client requests.

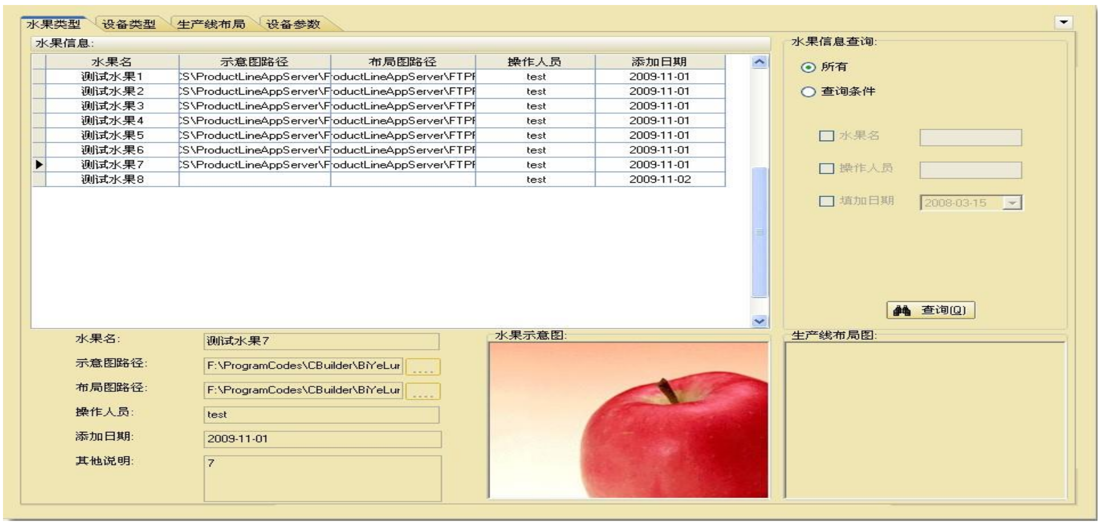
3.2. FTP Server

For real situations, the client may want to use the newest model files of the production line, requiring the FTP model download service from the server. In this paper, we adopt the Serv-U software to construct our FTP server. As a subdirectory of the server, the root directory contains five subdirectories, which are a background pictures subdirectory CS_Graph for login interfaces of client application and server application, a EON model files subdirectory Device_Model for production line devices, a subdirectory Fruit_Glass_Graph for pictures of fruits and containers, a layout graph subdirectory Layout_Graph for the production line, and a process graph subdirectory ProcessGraph for system applications.

3.3. Database Management

C++ Builder has provided a total of four database management standards, which are the Borland Database Engine (BDE), ActiveX Data Object (ADO), dbExpress, and InterBase Express. In this paper, we use the ADO standard duo to its simplicity.

Furthermore, we develop a convenient interface, as shown in Figure 3, to provide database management operations, such as database initialization and daily update management, of the fruit types table, the device parameters table and the production line layout table.



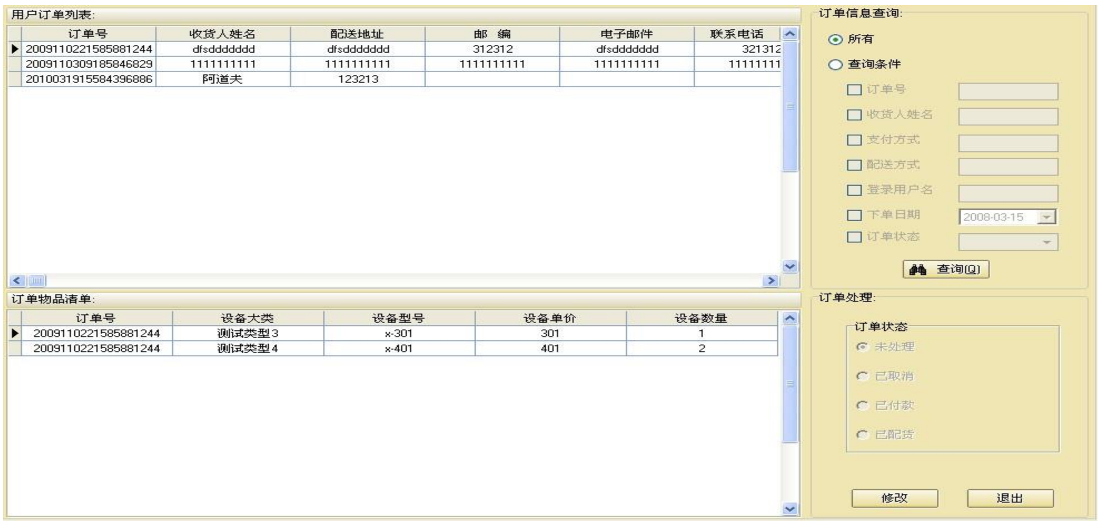
	水果名	示意图路径	布局图路径	操作人员	添加日期
	测试水果1	CS\ProductLineAppServer\F	oductLineAppServer\FTPF	test	2009-11-01
	测试水果2	CS\ProductLineAppServer\F	oductLineAppServer\FTPF	test	2009-11-01
	测试水果3	CS\ProductLineAppServer\F	oductLineAppServer\FTPF	test	2009-11-01
	测试水果4	CS\ProductLineAppServer\F	oductLineAppServer\FTPF	test	2009-11-01
	测试水果5	CS\ProductLineAppServer\F	oductLineAppServer\FTPF	test	2009-11-01
	测试水果6	CS\ProductLineAppServer\F	oductLineAppServer\FTPF	test	2009-11-01
▶	测试水果7	CS\ProductLineAppServer\F	oductLineAppServer\FTPF	test	2009-11-01
	测试水果8			test	2009-11-02

水果名:	测试水果7
示意图路径:	F:\ProgramCodes\CBuilder\BiYeLur
布局图路径:	F:\ProgramCodes\CBuilder\BiYeLur
操作人员:	test
添加日期:	2009-11-01
其他说明:	7

Figure 3. Database Management

3.4. Order Management

For order management, we also develop an intuitive interface, as shown in Figure 4, to manage the order statuses, where we use 0-processing, 1-paid and 2-cancelled, and query required items by SQL Select sentences.



用户订单列表:

订单号	收货人姓名	配送地址	邮 编	电子邮件	联系电话
2009110221585881244	dfsdddddd	dfsdddddd	312312	dfsdddddd	321312
2009110309185846829	1111111111	1111111111	1111111111	1111111111	11111111
2010031915584396886	阿道夫	123213			

订单物品清单:

订单号	设备大类	设备型号	设备单价	设备数量
2009110221585881244	测试类型3	x-301	301	1
2009110221585881244	测试类型4	x-401	401	2

订单信息查询:

[✓ 所有] 查询条件

订单号
收货人姓名
支付方式
配送方式
登录用户名
下单日期 2008-03-15
订单状态

查询(Q)

订单处理:

订单状态

[✓ 未处理] 已取消 已付款 已配货

修改 退出

Figure 4. Order management

4. Client Implementation

4.1. Customizable Configuration

4.1.1. Key Algorithms: One of the major contributions of the paper is the implementation of customizable configuration of the juicing and filling production line for our customers, with steps:

(1) Create a new project, by click the menu File-->NewApplication on the C++ Builder integrated development environment (IDE).

(2) Add related controls to the form editor of the new project, such as the ADOConnection control which is used to connect the database, and the ADOQuery control which is used to query related tables.

(3) Connect to the database and read the device list. When customers enter the device selection system, details of all devices in the production line are displayed in tree structure. Specifically, connect to the database and select devices from SZSCX_DeviceParam table with Tag_ProLine=0 for juicing production line. The codes are as follows.

```
    m_pAdo->Connection = acLeMin;
    m_pAdo->SQL->Text = "Select * from SZSCX_DeviceParam where Tag_ProLine =
0";
    m_pAdo->Active = true;
    ReadAndDisplay();
```

Then, we define a ReadAndDisplay() function to read and display the selected devices. In the ReadAndDisplay() function, we declare firstly the root node and temporary node, and then add device details into the left tree structure by using if and for sentences. The codes of the ReadAndDisplay() function are as follows.

```
    rootNode = rtvMenuTree->Items->Item[0];
    int count = m_pAdo->RecordCount;
    if (m_pAdo->RecordCount != 0)
        for (int i=1; i<=m_pAdo->RecordCount; ++i)
        {
                tempNode = rtvMenuTree->Items->AddChild(rootNode,m_pAdo->
                            FieldByName("Device_Type")->AsString);
                tempNode = rtvMenuTree->Items->AddChild(tempNode,m_pAdo->
                            FieldByName("Device_ModelNum")->AsString);
            }
```

After this, we select devices from SZSCX_DeviceParam table with Tag_ProLine=1 for filling production line. Instead of the ReadAndDisplay() function, we define a GoAndMakeTree() function to connect to the SZSCX_FruitDeviceLayout table and add the device layout into the right tree structure.

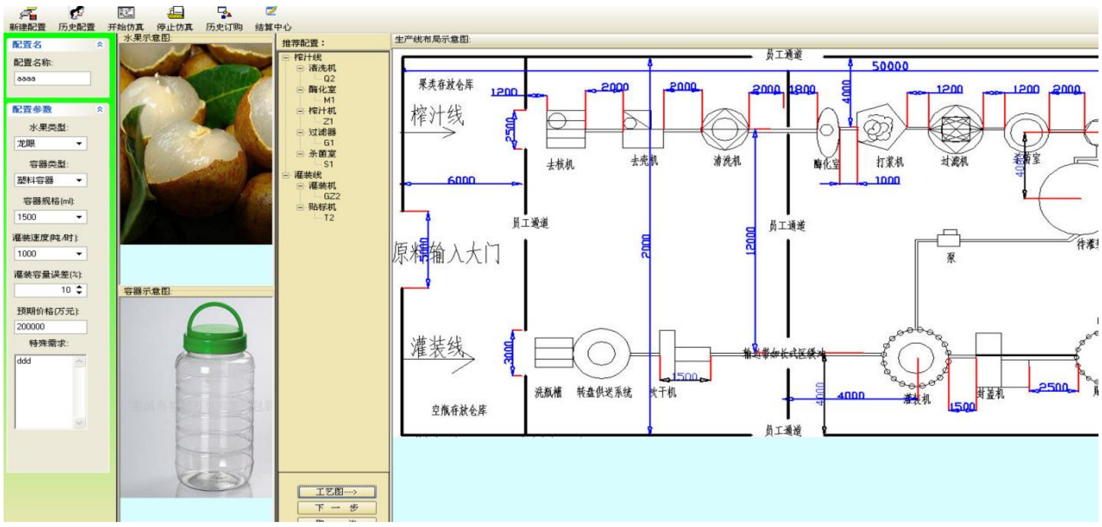
(4) Select the destination device and display its parameters and 3D EON model. When customers expend the above left tree structure, names and types of all devices are displayed. Besides, we use the following codes to display details of the selected devices in the center of the client interface, when customers click and select the required device.

```
    selectedNode = rtvMenuTree->Selected;
    if (selectedNode)
    {
        m_pAdo->SQL->Text = "Select * from SZSCX_DeviceParam where
                                Device_ModelNum = '"+ selectedNode->Text + "'";
        RzEdit->Text = m_pAdo->FieldByName("Device_Type")->AsString;
    }
```

(5) Add selected devices to the right layout tree structure and save the current configuration. When the customers are satisfied with the parameters and the 3D model of the selected devices, they can add them to the current configuration by the on click event function, as

```
    selectedNode = rtvMenuTree->Selected
    for (int i=0; i<rtvLineLayout->Items->Count; ++i)
    {
        if          (selectedNode->Parent->Text!="Juicing"||selectedNode->Parent-
>Text!="Filling")
        if (selectedNode->Parent->Text == rtvLineLayout->Items->Item[i]->Text)
            tempNode2=rtvLineLayout->Items->AddChild(  rtvLineLayout->Items-
                              >
                                            Item[i],selectedNode-
                                  >Text);
    }
```

4.1.2. Implemented Interfaces: When customers log into the client simulation system and create a new configuration, a requirement configuration interface pops up, as shown in Figure 5. Firstly, the customer should type in a text format configuration name, choose the fruit type and the container type from paper container, metal container, glass container and plastic container. Secondly, the customer should input the container volume, filling speed, filling error and total expected price. Last, the customer either confirms or cancels this configuration.



配置名

配置名称: aaaa

配置参数

水果类型: 龙眼

容器类型: 塑料容器

容器规格(ml): 1500

灌装速度(吨/时): 1000

灌装容量误差(%): 10

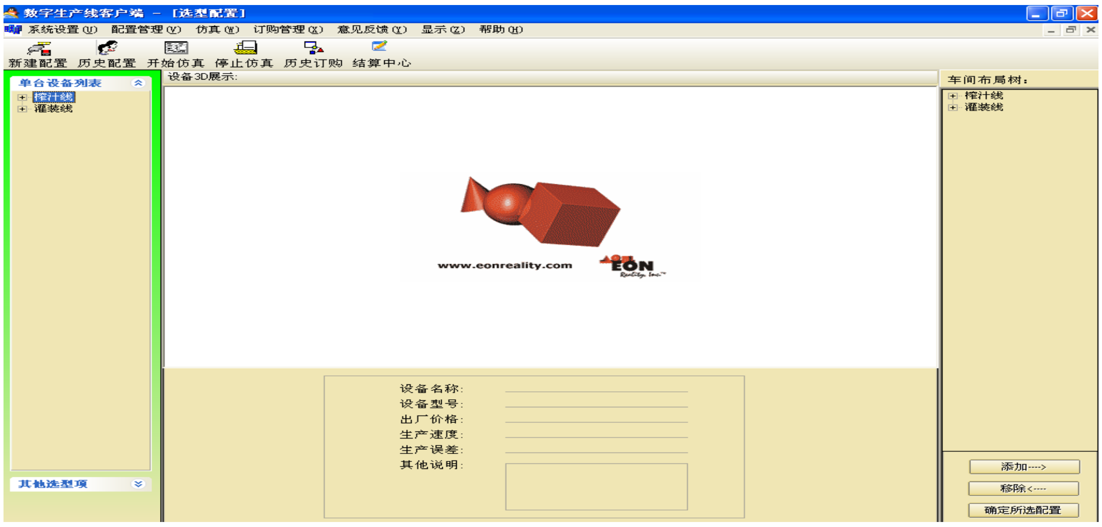
预期价格(万元): 200000

特殊需求: ddd

推荐配置：
```
⊟ 榨汁线
  ⊟ 清洗机
      Q2
  ⊟ 酶化室
      M1
  ⊟ 榨汁机
      Z1
  ⊟ 过滤器
      G1
  ⊟ 杀菌室
      S1
⊟ 灌装线
  ⊟ 灌装机
      GZ2
  ⊟ 贴标机
      T2
```

Figure 5. Requirement Configuration Interface

After the requirement configuration, the customer could enter in the device type selection interface, as shown in Figure 6. The left part of the figure shows the left tree structure which lists all the devices that meet the requirement configuration. When the customer clicks one device from the left tree structure, detailed parameters, such as name, type, price and error, of this selected device are displayed in the middle bottom of the interface, and its EON model will be displayed in the middle top of the interface.



设备名称:
设备型号:
出厂价格:
生产速度:
生产误差:
其他说明:

Figure 6. Device Type Selection Interface

4.2. Interactive Control

4.2.1. Key Algorithms: The other major contribution of the paper is the implementation of interactive control of the juicing and filling production line for our customers, which involves three aspects, as:

(1) Use the EON control to display dynamically the 3D device model in the C++Builder application, as shown in the middle top of the Figure 6. Due to the limit space, we list only the key codes, as follows, where the FileExists() function is used to search the EON model file in the specified directory, according to the device name.

```
if (!selectedNode->HasChildren)
```

```
{
    if (!m_strEonFile.IsEmpty())
    ctrEon->Stop();
    m_strEonFile = selectedNode->Text + ".Eoz";
    if(FileExists(ExtractFilePath(Application->ExeName) + "DeviceModel\\"
                                + m_strEonFile ))
        ctrEon->SimulationFile = ExtractFilePath(Application->ExeName)
                                + "DeviceModel\\"  + m_strEonFile;
    ctrEon->Start();
}
```

(2) Use the message passing mechanism to control the displayed 3D EON model. Since we start already the EON model in C++ Builder applications, the message passing mechanism of the EONX control, which encapsulates a large number of properties and methods, could be used to provide interactive simulations. Then, C++ Builder applications could communicate with EONX control by carrying out the following procedures, based on the provided methods. Firstly, create nodes of InEvent or OutEvent in the EON studio. Secondly, connect related InEvent nodes or OutEvent nodes, which is also called the route configuration of the EON studio platform. Last, add the OnEvent function of EONX control to C++Builder and monitor events in the InEvent function of EONX control. We list also only the key codes, as follows,

```
void __fastcall TfrmNewConfig::ctrEonEvent(TObject *Sender,
   BSTR bstrNodeName, Variant *pvarNodeValue)    // Monitor events
{
    if(bstrNodeName = = “OutEvent”)
    {
        //…Monitoring and processing of OutEvent nodes of the EON studio
platform
    }
}
```

(3) Show customers quotation results for their specified configurations. Firstly, computer the total filling speed, which is determined by the minimum speed of the devices in current configurations. The codes are:

```
int Speed = 0;
int m = m_pAdo->RecordCount;
if(i==0||speed>StrToInt(m_pAdo->FieldByName("Produce_Speed")->AsString))
    speed=StrToInt(m_pAdo->FieldByName("Produce_Speed")->AsString);
RzEdit32->Text = IntToStr(speed);
```

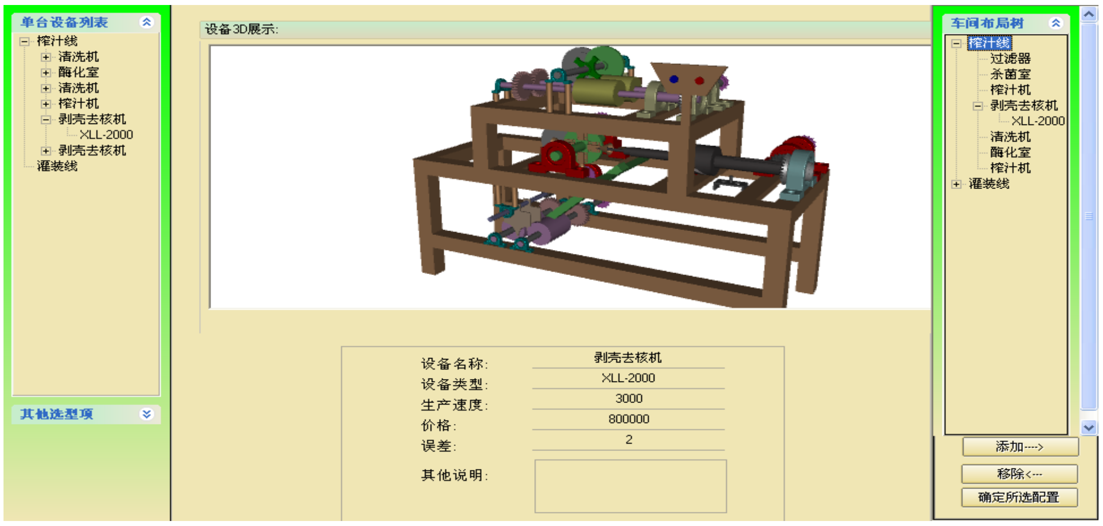
Secondly, compute the filling error, which is the sum of errors of all the devices n current configurations. The codes are:

```
int Error = 0;
int m = m_pAdo->RecordCount;
ListItem->SubItems->Add(m_pAdo->FieldByName("Produce_Error ")->AsString);
Error+= StrToInt(m_pAdo->FieldByName("Produce_Error ")->AsString);
RzEdit33->Text = IntToStr( Error);
```

Last, compute the total price, which is the sum of prices of all he devices n current configurations. The codes are:

```
int price = 0;
int m = m_pAdo->RecordCount;
ListItem->SubItems->Add(m_pAdo->FieldByName("Device_Price")->AsString);
price+= StrToInt(m_pAdo->FieldByName("Device_Price")->AsString);
RzEdit25->Text = IntToStr( price);
```

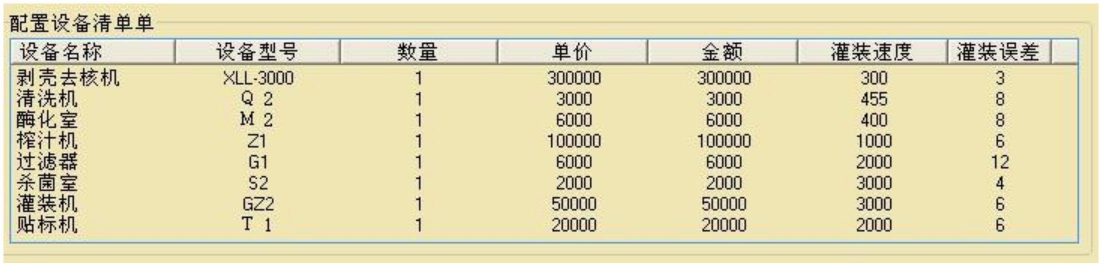
4.2.2. Implemented Interfaces: Figure 7 shows the layout result of the production line by interactive controls on above device type selection interface, where the right tree structure is generated from the left tree structure by current configurations.



设备名称: 剥壳去核机
设备类型: XLL-2000
生产速度: 3000
价格: 800000
误差: 2
其他说明:

Figure 7. Add Devices to the Layout of the Production Line

It can be seen from this Figure 7 that the cracking and pitting machine XLL-2000 is already in the right layout tree. Once we select all needed devices from the left tree structure, we can then click the OK button to save current configuration and enter the quotation results interface, as shown in Figure 8.



配置设备清单单

设备名称	设备型号	数量	单价	金额	灌装速度	灌装误差
剥壳去核机	XLL-3000	1	300000	300000	300	3
清洗机	Q 2	1	3000	3000	455	8
酶化室	M 2	1	6000	6000	400	8
榨汁机	Z1	1	100000	100000	1000	6
过滤器	G1	1	6000	6000	2000	12
杀菌室	S2	1	2000	2000	3000	4
灌装机	GZ2	1	50000	50000	3000	6
贴标机	T 1	1	20000	20000	2000	6

Figure 8. Quotation Results Interface

5. Conclusion

In this paper, we design and implement a client-server architecture based simulation system for the juicing and filling production line, where we describe requirements analysis and database design in the system analysis and design section, the application server, the FTP server, the database management and the order management in the server implementation section, the customizable configuration and the interactive control in the client implementation section. We also apply our simulation system to practical applications. The results show the effectiveness of our new approach for the design of the juicing and filling production line.

Acknowledgments

This work was supported by the China National Science Foundation under Grants (61201162, 61402106), Basic Research Program of Jiangsu Province (NSF), China (BK2012434), Postdoctoral Research Fund of China (2012M511790), Natural Science Foundation of the Higher Education Institutions of Jiangsu Province, China (12KJB510020), New Teacher Fund for Doctor Station of the Ministry of Education, China (20123223120001) and the Doctor Startup Foundation of Wuyi University under Grant No. 2014BS07. We would like to thank anonymous reviewers for helpful comments.

References

[1] P. H. Tsarouhas and I. S. Arvanitoyannis, “Yogurt production line: reliability analysis,” *Prod. Manuf. Res.*, vol. 2, no. 1, **(2014)**, pp. 11-23.
[2] J. L. Zhang, A. R. Zhang, and Z. L. Zhang, “The realization of a mixed automatic production line based on PLC,” In: *Appl. Mech. Mater.*, vol. 610, **(2014)**, pp. 506-509.
[3] H. Wang, et al., “Study on behavior simulation for picking manipulator in virtual environment based on binocular stereo vision,” In *ICSC*, **(2008)**, pp. 27-31.
[4] G. Rudolf, N. Noyan, and V. Giard, “Modeling sequence scrambling and related phenomena in mixed-model production lines,” *Eur. J. Oper. Res.*, vol. 237, no. 1, **(2014)**, pp. 177-195.
[5] H. Li, et al., “Catalytic hydrothermal pretreatment of corncob into xylose and furfural via solid acid catalyst,” *Bioresource Technol.*, vol. 158, **(2014)**, pp. 313-320.
[6] Y. Liu, X. Luo, and Z. Li, “Microstructure evolution during semi-solid powder rolling and post-treatment of 7050 aluminum alloy strips,” *J. Mater. Proc. Tech.*, vol. 214, no. 2, **(2014)**, pp. 165-174.
[7] J. X. Chen, et al., “Modeling and performance analyzing of helix transmission base on Modelica,” *Key Eng. Mater.*, vol. 455, **(2011)**, pp. 511-515.
[8] Z. M. Bi and B. Kang, “Sensing and responding to the changes of geometric surfaces in flexible manufacturing and assembly,” *Enterp. Inform. Syst.*, vol. 8, no. 2, **(2014)**, pp. 225-245.
[9] C. Liu, et al., “Research on service-oriented and HLA-based simulation model of juice production line,” In *ICMTMA*, vol. 3, **(2010)**, pp. 167-170.
[10] P. Greschke, M. Schönemann, S. Thiede, and C. Herrmann, “Matrix structures for high volumes and flexibility in production systems,” *Procedia CIRP*, vol. 17, **(2014)**, pp. 160-165.
[11] L. Chen, X. Wang, and L. Tang, “Operation optimization in the hot-rolling production process,” *Ind. Eng. Chem. Res.*, vol. 53, no. 28, **(2014)**, pp. 11393-11410.
[12] H. L. Li, et al., “One-step heterogeneous catalytic process for the dehydration of xylan into furfural,” *BioResources*, vol. 8, no. 3, **(2013)**, pp. 3200-3211.
[13] Z. N. Kain, et al., “The perioperative surgical home as a future perioperative practice model,” *Anesth. Analg.*, vol. 118, no. 5, **(2014)**, pp. 1126-1130.
[14] B. Lu, et al., “Discovery of community structure in complex networks based on resistance distance and center nodes,” *J. Comput. Inf. Syst.*, vol. 8, no. 23, **(2012)**, pp. 9807-9814.
[15] T. M. Deboni, M. Bündchen, C. V. Junior, D. Hotza, R. Piletti, and M. G. N. Quadri, “Effect of the processing steps on cactus juice production,” *Food Bioprocess Tech*., vol. 7, no. 4, **(2014)**, pp. 990-1000.
[16] M. V. P. Rocha, et al., “Biosurfactant production by Pseudomonas aeruginosa MSIC02 in cashew apple juice using a 24 full factorial experimental design,” *Chem. Ind. Chem. Eng. Q.*, vol. 20, no. 1, **(2014)**, pp. 49-58.
[17] C. Y. Liu, et al., “Study on adaptive and fuzzy weighted image fusion based on wavelet transform in trinocular vision of picking robot,” *J. Inf. Comput. Sci.*, vol. 11, no. 6, **(2014)**, pp. 1929-1937.
[18] H. Ni, et al., “Pectinase and naringinase help to improve juice production and quality from pummelo (Citrus grandis) fruit,” *Food Sci. Biotech.*, vol. 23, no. 3, **(2014)**, pp. 739-746.
[19] H. J. Wang, et al., “Study on a location method for bio-objects in virtual environment based on neural network and fuzzy reasoning,” In *ICIRA*, vol. 5928, **(2009)**, pp. 1004-1012.
[20] K. H. Choi, H. Lee, and Y. Kim, “A preliminary study on energy saving smart space using location awareness technology,” *Int. J. Contr. Autom.*, vol. 7, no. 6, **(2014)**, pp. 153-158.
[21] M. Mosleh, H. Noori, and A. Nikdel, “Cluster based and energy efficient coverage protocol for wireless sensor networks,” *Int. J. Smart Home*, vol. 8, no. 2, **(2014)**, pp. 39-54.

Authors

Changyu Liu, he joined the Communication and Computer Network Lab of Guangdong as a PhD student in 2010 at South China University of Technology, advised by Prof. Shoubin Dong. He was a Visiting Scholar at the School of Computer Science, Carnegie Mellon University, from September 2012 to October 2013, advised by Dr. Alex Hauptmann. Then, he worked with Prof. Mohamed Abdel-Mottaleb at the Department of Electrical and Computer Engineering, University of Miami, from October 2013 to September 2014. His research interests include computer vision and machine learning.

Dapeng Li, he received the B.S. degree in Electronics from Harbin Engineering University, China, in 2003, and the M.S. degree in Communication Systems from Harbin Engineering University, China, in 2006. He received his Ph.D. degree from the Department of Electronic Engineering, Shanghai Jiao Tong University, China, in 2010. From Oct. 2007 to Aug. 2008, he have joined in the 4G/LTE development at the R&D Center, Huawei Company, Shanghai, China. Starting Jan. 2011, he is a faculty member in the College of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications, China. From 03/14-03/15, he is a visiting scholar at University of Miami and Virginia Tech. His research interests include future cellular networks, machine to machine communications, mobile ad hoc networks, cognitive radio networks, radio resource management and cooperative communications.

Bin Lu, he is currently a lecturer in the School of Computer Science at Wuyi University. He received his Ph.D. degree in 2013 from South China University of Technology. He is a reviewer of the Journal of Yangtze River Scientific Research Institute since 2010. His main research interests include complex network and machine learning.

Tiezhu Zhao, he received the Ph.D. degree from South China University of Technology in 2011. He is currently a research associate at the Computer College, Dongguan University of Technology. His research interests include distributed/parallel computing, pattern recognition and network security.