Applied Mathematics and Nonlinear Sciences

<https://www.sciendo.com>

Research on Simufact simulation data processing system based on QT and MySQL

Liang Zhao^{1,†}, Hengzhi Gu², Jingxian Xu², Yang Cui², Chunjiang Shuai³

¹ School of Mechanical Engineering, Shaanxi University of Technology, Hanzhong Shaanxi, China

² School of Automobile, Chang'an University, Xi'an Shaanxi, China

³ School of Physics and Telecommunications Engineering, Shaanxi University of Technology, Hanzhong Shaanxi, China

Submission Info

Communicated by Juan Luis García Guirao

Received February 25th 2021

Accepted July 22nd 2021

Available online November 10th 2021

Abstract

As the simulation data of the Simufact Additive software was inconvenient to process, QT and structured query language (MySQL) database were used to develop the database to solve the processing of the simulation data. The GUI interface that has the functions of processing and saving the simulation data was designed. The interface developed by QT was used to start the Simufact Additive software, connect to the MySQL database and save the simulation data. The results showed that using MySQL as the management of simulation data has achieved better results in the post-processing of simulation data.

Keywords: QT, MySQL, Simufact Additive

1 Introduction

Simufact Additive software is developed by MSC software company for additive manufacturing simulation, and its software interface is shown in Figure 1. The software provides a simple user interface that can help users to quickly establish the parts model of additive manufacturing and also support the simulation of the post-treatment process of additive manufacturing (such as heat treatment or hot isostatic pressing). The software can be used for numerical calculation of additive manufacturing process of wire or powder materials under laser, arc or electron beam heat source. Good results can also be obtained for SLM, direct metal sintering (DMLS), electron beam melting (EBM) or laser casting [1].

Simufact Additive software is a simulation software specially developed for the metal additive manufacturing process. It can evaluate the structure design, material selection, manufacturing parameters, stacking forming

[†]Corresponding author.

Fig. 1 Interface of Simufact software.

direction and support structure of the additive manufacturing parts, and carry out the combination of process parameters and design structure and optimization. When using a 3D printer to make a metal model, the 3D printing process can be simulated through the user interface and the simulation data can be saved in a data file, but it is not convenient to operate when the simulation data is processed in the later stage [2]. Therefore, it is necessary to develop a processing system for Simufact Additive simulation data. In this paper, QT and MySQL databases are used to develop Simufact Additive simulation data processing. QT is a cross-platform C++ graphical user interface application-development framework developed by Qt Company in 1991. It can be used to develop GUI programs as well as non-GUI programs. MySQL is a relational database management system. Relational databases store data in different tables, thus increasing the speed of data access and flexibility [3–5].

2 Simufact Additive simulation data

The postprocessing function of Simufact can be displayed in many forms such as text files, pictures, animation, curves, etc. And it can be sliced to display various fields inside the workpiece. It can display material flow lines, variable point queries at any point and tracing of particle flow, and it allows multiple windows to display so that the results can be compared [6].

When using Simufact Additive software for simulation, three files will be produced after the simulation is completed, namely Log file, Status file and Output file. The status data during the simulation process is saved in the Status file. This is a notepad file. The status data during the 3D printing simulation process is recorded in detail in the file, but it is not convenient for later data processing.

Fig. 2 Display of Simufact simulation results.

Fig. 3 Simulation output of Simufact Additive.

3 MySQL database

3.1 Introduction to MySQL database

MySQL database is an open source relational database. It usually uses structured query language (SQL) to manage a variety of data or services. It has high performance, small size, portability and ease of use, which can satisfy different enterprises The business data storage requirements of the unit.

Similar to the conventional scripting languages, MySQL also has a set of rules for the use of characters,

words and special symbols. MySQL completes the operation of the database by executing SQL scripts. The script consists of one or more MySQL statements (SQL statements + extended statements), and the script file extension is generally .sql when saving. Under the console, the MySQL client can also execute single-sentence statements without saving them as .sql files.

MySQL database has the advantages of rich functions, speed and security. MySQL supports most of the functions that a relational database should have – either directly or indirectly. Moreover, it also supports many security features and some of them are very advanced and built into MySQL. Abandoning certain standards allows MySQL to work very efficiently and simply, and therefore in a faster manner. The most important thing is MySQL can handle large amounts of data and be used on a large scale when needed. That is exactly what we need.

3.2 Creating Simufact simulation database

After installing the MySQL database on the computer, create a database of Simufact simulation data and create a table in the Simufact database named “layerstatus (nu,case_name,case_nu,time,cyc,conv_r,conv_d,prg,pst,rst)” and “cuttingstatus (nu,case_name,time,cyc,conv_r,conv_d,prg,pst,rst)”.

Field	Type	Null	Key	Default	Extra
nu	int(11)	YES		NULL	
case_name	varchar(30)	YES		NULL	
case_nu	int(11)	YES		NULL	
time	varchar(30)	YES		NULL	
cyc	int(11)	YES		NULL	
conv_r	varchar(30)	YES		NULL	
conv_d	varchar(30)	YES		NULL	
prg	varchar(30)	YES		NULL	
pst	int(11)	YES		NULL	
rst	int(11)	YES		NULL	

Fig. 4 Table layerstatus.

Field	Type	Null	Key	Default	Extra
nu	int(11)	YES		NULL	
case_name	varchar(30)	YES		NULL	
time	varchar(30)	YES		NULL	
cyc	int(11)	YES		NULL	
conv_r	varchar(30)	YES		NULL	
conv_d	varchar(30)	YES		NULL	
prg	varchar(30)	YES		NULL	
pst	int(11)	YES		NULL	
rst	int(11)	YES		NULL	

Fig. 5 Table cuttingstatus.

4 Operations on the database in QT

4.1 Introduction to QT

Qt is a cross-platform C++ application framework that supports Windows, Linux, Mac OS X, Android, iOS, Windows Phone, embedded systems, etc. In other words, Qt can support desktop application development,

embedded development and mobile development at the same time, covering all existing mainstream platforms. You only need to write the code once and recompile it before publishing it to a different platform [7].

Qt is not just a GUI library. In addition to creating beautiful interfaces, it also has many other components. For example, you no longer need to study STL, C++ <string> header files, to find and parse XML, various third-party libraries for connecting to databases and Socket, and Qt have these built-in [8].

The interface source code is separated from the user source code. Qt's strategy is to generate a corresponding source code if there is no interface. For example, there is an interface design file `xps.ui` that will parse the XML and generate a corresponding `ui_xps.h` file. You do not need this file. Modify, if you want to call the controls in the interface directly through `Ui::XPSClass ui`; this object definition is also written in `ui_xps.h`. That is, your business logic is isolated from the creation and setting of interface control objects [9].

4.2 Connect to MySQL database in QT

We have installed MySQL database outside of QT platform and then created data table structure and added table database through CMD command. This method is intuitive to operate the database table, but it needs to install the program.

SQL statements are supported in QT. There are two ways to create database tables: one is to use MySQL common lineclient to execute SQL statements; the other is to use SQL model class to provide MVC mode with QT, such as `qsqlquerymodel` and `qsqltablemodel` class for database operation, and the results can be displayed in the table [10–12].

In order to deal with the simulation data of Simufact Additive software conveniently, we need to operate the database in MySQL in QT. We have already installed and configured this database. The next thing we need to do is to connect the database in QT. In order to achieve this goal, we write a program to connect the host, port number and name of the database in QT. Then enter the user name and password of the database to open the database. The process of connecting to the database is shown in Figure 6.

Fig. 6 QT connection to MySQL database process.

Through the program design, after the connection is successful, a dialogue box for connecting to the database will pop up, as shown in Figure 7.

4.3 Processing of simulation data in QT

Before saving the simulation data in the database, the program developed by QT needs to process the data in the data file simulated by the Simufact Additive software, delete irrelevant information in the file, and leave

Fig. 7 Successful database connection.

only one space between the data in the notepad file in order to save in the database. The data in the status data file is shown in Figure 8.

```
01-autoGenSupports.sts - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
information summary of sfMarc am1.0 rev. 7860 based on MARC 2015.1-cl.389571
job name : 01-autoGenSupports

inc load_case_name tot_time cyc conv_r conv_d prg(%) pst rst
-----
0 Initializing 0.0000E+00 0 0.000E+00 0.000E+00 10.00 0 0
0 Initializing 0.0000E+00 0 0.000E+00 0.000E+00 40.00 0 0
0 Initializing 0.0000E+00 0 0.000E+00 0.000E+00 80.00 0 0
0 Initializing 0.0000E+00 0 0.000E+00 0.000E+00 100.00 0 0
0 Start simulation 0.0000E+00 0 0.000E+00 0.000E+00 0.00 0 0
1 vxlayer 5 1.0000E+00 6 7.319E-02 0.000E+00 2.22 1 0
2 vxlayer 6 2.0000E+00 6 8.183E-02 0.000E+00 4.44 2 0
3 vxlayer 7 3.0000E+00 4 7.835E-02 0.000E+00 6.67 3 0
4 vxlayer 8 4.0000E+00 4 7.205E-02 0.000E+00 8.89 4 0
5 vxlayer 9 5.0000E+00 4 9.481E-02 0.000E+00 11.11 5 0
6 vxlayer 10 6.0000E+00 4 4.746E-02 0.000E+00 13.33 6 0
7 vxlayer 11 7.0000E+00 3 8.928E-02 0.000E+00 15.56 7 0
8 vxlayer 12 8.0000E+00 3 8.790E-02 0.000E+00 17.78 8 0
9 vxlayer 13 9.0000E+00 3 8.707E-02 0.000E+00 20.00 9 0
10 vxlayer 14 1.0000E+01 3 8.329E-02 0.000E+00 22.22 10 0
```

Fig. 8 Raw data in the status file.

Use the `readLine()` statement in QT to read the simulation data in the interface developed by QT. During the reading process, the blanks in the file are processed. The processing statement `data=data.append(in.readLine()).simplified()).append("\n")` displays the read data in the window developed by QT, as shown in Figure 9, deletes the invalid data in the file in the window interface and saves it in the newly created status data file, as shown in Figure 10.

```
information summary of sfMarc am1.0 rev. 7860 based on MARC 2015.1-cl.389571
job name : 01-autoGenSupports

inc load_case_name tot_time cyc conv_r conv_d prg(%) pst rst
-----
0 Initializing 0.0000E+00 0 0.000E+00 0.000E+00 10.00 0 0
0 Initializing 0.0000E+00 0 0.000E+00 0.000E+00 40.00 0 0
0 Initializing 0.0000E+00 0 0.000E+00 0.000E+00 80.00 0 0
0 Initializing 0.0000E+00 0 0.000E+00 0.000E+00 100.00 0 0
0 Start simulation 0.0000E+00 0 0.000E+00 0.000E+00 0.00 0 0
1 vxlayer 5 1.0000E+00 6 7.319E-02 0.000E+00 2.22 1 0
2 vxlayer 6 2.0000E+00 6 8.183E-02 0.000E+00 4.44 2 0
3 vxlayer 7 3.0000E+00 4 7.835E-02 0.000E+00 6.67 3 0
4 vxlayer 8 4.0000E+00 4 7.205E-02 0.000E+00 8.89 4 0
5 vxlayer 9 5.0000E+00 4 9.481E-02 0.000E+00 11.11 5 0
6 vxlayer 10 6.0000E+00 4 4.746E-02 0.000E+00 13.33 6 0
7 vxlayer 11 7.0000E+00 3 8.928E-02 0.000E+00 15.56 7 0
8 vxlayer 12 8.0000E+00 3 8.790E-02 0.000E+00 17.78 8 0
9 vxlayer 13 9.0000E+00 3 8.707E-02 0.000E+00 20.00 9 0
10 vxlayer 14 1.0000E+01 3 8.329E-02 0.000E+00 22.22 10 0
```

Fig. 9 QT reads the middle data of the status file.

Use the `QsqlQuery` class to provide SQL statement operations and save the newly created status data file in a table created in the MySQL database.

`QsqlQuery query;`

layerstatus - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```

1 vxlayer 5 1.0000E+00 6 7.319E-02 0.000E+00 2.22 1 0
2 vxlayer 6 2.0000E+00 6 8.183E-02 0.000E+00 4.44 2 0
3 vxlayer 7 3.0000E+00 4 7.835E-02 0.000E+00 6.67 3 0
4 vxlayer 8 4.0000E+00 4 7.205E-02 0.000E+00 8.89 4 0
5 vxlayer 9 5.0000E+00 4 9.481E-02 0.000E+00 11.11 5 0
6 vxlayer 10 6.0000E+00 4 4.746E-02 0.000E+00 13.33 6 0
7 vxlayer 11 7.0000E+00 3 8.928E-02 0.000E+00 15.56 7 0
8 vxlayer 12 8.0000E+00 3 8.790E-02 0.000E+00 17.78 8 0
9 vxlayer 13 9.0000E+00 3 8.707E-02 0.000E+00 20.00 9 0
10 vxlayer 14 1.0000E+01 3 8.329E-02 0.000E+00 22.22 10 0

```

Fig. 10 Data after QT processing.

mysql> select *from layerstatus;									
nu	case_name	case_nu	time	cyc	conv_r	conv_d	prg	pst	rst
1	vxlayer	5	1.0000E+00	6	7.319E-02	0.000E+00	2.22	1	0
2	vxlayer	6	2.0000E+00	6	8.183E-02	0.000E+00	4.44	2	0
3	vxlayer	7	3.0000E+00	4	7.835E-02	0.000E+00	6.67	3	0
4	vxlayer	8	4.0000E+00	4	7.205E-02	0.000E+00	8.89	4	0
5	vxlayer	9	5.0000E+00	4	9.481E-02	0.000E+00	11.11	5	0
6	vxlayer	10	6.0000E+00	4	4.746E-02	0.000E+00	13.33	6	0
7	vxlayer	11	7.0000E+00	3	8.928E-02	0.000E+00	15.56	7	0
8	vxlayer	12	8.0000E+00	3	8.790E-02	0.000E+00	17.78	8	0
9	vxlayer	13	9.0000E+00	3	8.707E-02	0.000E+00	20.00	9	0
10	vxlayer	14	1.0000E+01	3	8.329E-02	0.000E+00	22.22	10	0
11	vxlayer	15	1.1000E+01	3	9.764E-02	0.000E+00	24.44	11	0
12	vxlayer	16	1.2000E+01	3	8.401E-02	0.000E+00	26.67	12	0
13	vxlayer	17	1.3000E+01	3	9.100E-02	0.000E+00	28.89	13	0
14	vxlayer	18	1.4000E+01	3	8.363E-02	0.000E+00	31.11	14	0
15	vxlayer	19	1.5000E+01	3	7.522E-02	0.000E+00	33.33	15	0
16	vxlayer	20	1.6000E+01	3	7.824E-02	0.000E+00	35.56	16	0
17	vxlayer	21	1.7000E+01	3	8.512E-02	0.000E+00	37.78	17	0
18	vxlayer	22	1.8000E+01	3	7.230E-02	0.000E+00	40.00	18	0
19	vxlayer	23	1.9000E+01	3	6.105E-02	0.000E+00	42.22	19	0
20	vxlayer	24	2.0000E+01	3	6.017E-02	0.000E+00	44.44	20	0
21	vxlayer	25	2.1000E+01	3	5.009E-02	0.000E+00	46.67	21	0
22	vxlayer	26	2.2000E+01	3	5.631E-02	0.000E+00	48.89	22	0
23	vxlayer	27	2.3000E+01	3	5.766E-02	0.000E+00	51.11	23	0
24	vxlayer	28	2.4000E+01	2	9.383E-02	0.000E+00	53.33	24	0
25	vxlayer	29	2.5000E+01	2	5.801E-02	0.000E+00	55.56	25	0
26	vxlayer	30	2.6000E+01	3	5.033E-02	0.000E+00	57.78	26	0
27	vxlayer	31	2.7000E+01	2	9.749E-02	0.000E+00	60.00	27	0
28	vxlayer	32	2.8000E+01	2	9.995E-02	0.000E+00	62.22	28	0
29	vxlayer	33	2.9000E+01	3	4.717E-02	0.000E+00	64.44	29	0
30	vxlayer	34	3.0000E+01	2	9.674E-02	0.000E+00	66.67	30	0
31	vxlayer	35	3.1000E+01	2	9.456E-02	0.000E+00	68.89	31	0
32	vxlayer	36	3.2000E+01	2	9.026E-02	0.000E+00	71.11	32	0
33	vxlayer	37	3.3000E+01	2	9.126E-02	0.000E+00	73.33	33	0
34	vxlayer	38	3.4000E+01	3	3.849E-02	0.000E+00	75.56	34	0
35	vxlayer	39	3.5000E+01	3	5.363E-02	0.000E+00	77.78	35	0
36	vxlayer	40	3.6000E+01	3	5.852E-02	0.000E+00	80.00	36	0
37	vxlayer	41	3.7000E+01	3	6.572E-02	0.000E+00	82.22	37	0
38	vxlayer	42	3.8000E+01	3	6.683E-02	0.000E+00	84.44	38	0
39	vxlayer	43	3.9000E+01	3	3.929E-02	0.000E+00	86.67	39	0
40	vxlayer	44	4.0000E+01	3	4.055E-02	0.000E+00	88.89	40	0
41	vxlayer	45	4.1000E+01	3	4.082E-02	0.000E+00	91.11	41	0
42	vxlayer	46	4.2000E+01	3	4.341E-02	0.000E+00	93.33	42	0
43	vxlayer	47	4.3000E+01	3	4.159E-02	0.000E+00	95.56	43	0
44	vxlayer	48	4.4000E+01	3	4.218E-02	0.000E+00	97.78	44	0
45	vxlayer	49	4.5000E+01	2	9.568E-02	0.000E+00	100.00	45	0

Fig. 11 The database table data.

```

query.exec("load data local infile 'd:/layerdata.txt' into table layerstatus'
"character set Utf8 fields terminated by '"
"(nu,case_name,case_nu,time,cyc,conv_r,conv_d,prg,pst,rst)");

```

Use MySQL commands to display database table data as shown in Figure 11.

5 Conclusions

As the Simufact Additive software simulation data is inconvenient to process, this design uses QT and MySQL to realise the development of the database, which solves the storage function of the simulation data obtained in the Simufact Additive software simulation process in the MySQL database. The connection with the database is realised in QT, the SQL statement is used to create the table, and the SQL statement operation is provided through the QSqlQuery class to save the data in the database, which is convenient for processing the data in the database.

References

- [1] Hong Chang, Numerical simulation of forming stress and deformation of GH4169 powder SLM based on simulact. Shandong Jianzhu University, 2019.
- [2] Song Qingjun, Jing Qi, Zhang Wei. Simulation Research on 3D Printing of Selective Laser Melting of Automotive Aluminum Alloy Guide Arm. *Automotive technology and materials*, 2020, (10), pp. 47-53.
- [3] Han Ganning, Li Yongfeng, Gao Yiteng, Design and Development of MySQL Database Based on Embedded Qt. *Microcomputer application*, 2020, 36(05), pp. 25-27.
- [4] Pan Haoda, Zheng Peng, Zheng Liang, etc, Industrial Remote Control System Based on Qt and Web. *Radio communication technology*, 2015, 41(06), pp. 88-91+96.
- [5] Li Pengwei, Zhao Hanchao, Zhang Xinqi, etc, Implementation of a MySQL-based laser ranging data analysis platform. *Computer Era*, 2020, (07), pp. 37-39+42.
- [6] Wu Jianhua, Application of Simufact software in plastic machining. *Aviation Precision Manufacturing Technology*, 2010, (02), pp. 43-45.
- [7] Zhang, Han, Yin, Hongtao, Fu, Ping Design of Data Acquisition Software for Steam Turbine Based on Qt/Embedded. *Smart Innovation, Systems and Technologies*, 2020, 156, pp. 375-384.
- [8] Zhang Jianwei, Tian Xingke, Duan Qian, Design and implementation of vehicle terminal graphic interface based on QT. *Journal of Physics: Conference Series*, v 1486, n 7, April 17, 2020.
- [9] Du Haiming, Design of Vehicle Remote Monitoring System Based on LoRa Technology and QT Platform. *Communications in Computer and Information Science*, 2020, 1252, pp. 449-460.
- [10] Han Gaining, Li Yongfeng, Gao Yiteng, Design and Development of MySQL Database Based on Embedded Qt. *Microcomputer Applications*, 2020, 36(05), pp. 25-27.
- [11] Ni Jianyun, Xie Shuzhi, Li Zihao, Jia Chao, Optimization design method of cache extension in MySQL database. *Proceedings of 2018 IEEE International Conference on Mechatronics and Automation, ICMA 2018*, pp. 2295-2299.
- [12] Balabanova, Ivelina Stefanova, Georgiev, Georgi Ivanov, Kostadinova, Stela Savova, WEB Design of Digital Filters in LabVIEW and Connection with MSSQL and MySQL Databases. *2018 IEEE 27th International Scientific Conference Electronics, ET 2018 – Proceedings*.