УДК 681.5
DOI: 10.35211/1990-5297-2023-8-279-51-54

М. А. Волохов, А. В. Дроботов

АВТОМАТИЗИРОВАННАЯ СИСТЕМА ПОДГОТОВКИ 3D МОДЕЛЕЙ К ПРОИЗВОДСТВУ ИЗДЕЛИЙ НА АДДИТИВНЫХ УСТАНОВКАХ

Волгоградский государственный технический университет
E-mail: maxim_volohov@mail.ru

Рассматривается проблема длительности подготовки моделей к 3D печати с целью снижения трудоемкости процесса подготовки модели к производству на аддитивных установках и самого производства. Описаны подключение и обмен данными по сокету между сервером и движком CuraEngine. Сравнены результаты длительности подготовки 3D моделей к печати с учетом предлагаемой системы и без нее.

Ключевые слова: 3D принтер, автоматизация, 3D печать, обработка информации, слайсинг, проектирование моделей, сервер, сокет.

M. A. Volokhov, A. V. Drobotov

AUTOMATED SYSTEM FOR PREPARING 3D MODELS FOR THE PRODUCTION OF PRODUCTS ON ADDITIVE INSTALLATIONS

Volgograd State Technical University

The problem of the duration of the preparation of models for 3D printing is considered, in order to reduce the complexity of the process of preparing the model for production at additive plants and the production itself. Connection and data exchange via socket between the server and the CuraEngine engine are described. The results of the duration of preparation of 3D models for printing with and without the proposed system are compared.

Keywords: 3D printer, automation, 3D printing, information processing, slicing, model design, server, socket.

В последние годы аддитивные технологии все чаще используются в разных сферах промышленности, а иногда они и вовсе вытесняют традиционное единичное или мелкосерийное производство. Они получили такую популярность из-за низкого расхода материалов, создания геометрических форм практически любых размеров и неограниченной сложности, а также доступности производственного оборудования. Большого успеха аддитивное производство также достигло в отрасли быстрого прототипирования, позволяя оперативно создавать тестовые образцы изделий.

Однако у аддитивного производства также имеются недостатки: ограниченный размер изделия, высокая шероховатость поверхности после печати, затраты времени на подготовку модели к печати, потребность в высококвалифицированном персонале, в зависимости от материала – могут быть вредные выбросы [1]. Разработанная нами автоматизированная система для подготовки G-кода (программы производства изделия) по его 3D модели в формате STL, позволит уменьшить затраты времени на подготовку к печати, а также снизит потребность в персонале, настраивающем слайсер для подготовки G-кода.

© Волохов М. А., Дроботов А. В., 2023.

Рассмотрим процесс подготовки модели червячного вала к печати, который схематично представлен на рис. 1. Сначала червячный вал нужно спроектировать в САПР системе, после преобразовать в список треугольных граней, описывающих поверхность, и нарезать на слои, чтобы описать процесс печати в G-коде [2].

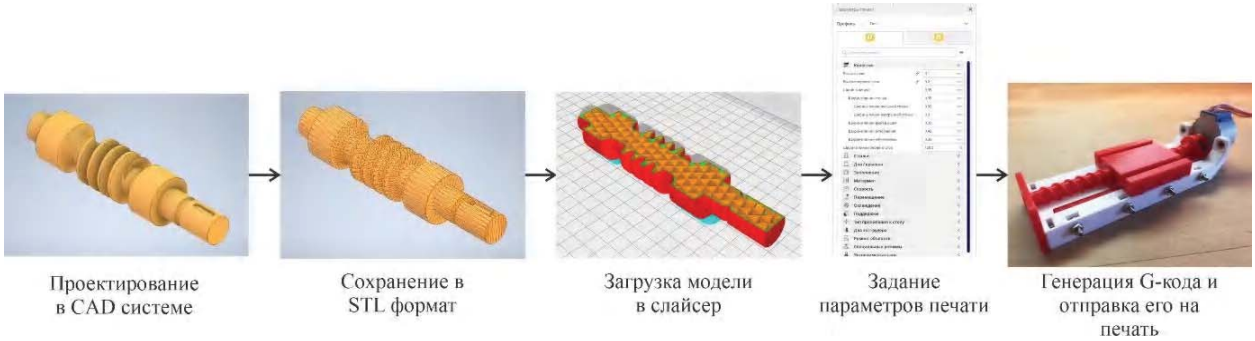

Рис. 1. Процесс подготовки модели к печати

Одной из самых популярных программ для 3D печати является UltiMaker Cura, позволяющая загружать STL модели, заполнять параметры печати, нарезать модель и генерировать G-код [0]. Cura – приложение-слайсер, в основе которого лежит движок CuraEngine, который как раз и выполняет все математические расчеты над моделью, создавая команды печати. Используя открытый исходный код движка и общаясь с ним напрямую, можно в несколько раз уменьшить время преобразования модели в инструкции G-кода. При таком методе не потребуется тратить время на визуализацию для пользователя и загрузку ненужных интерфейсных функций. На рис. 2 проиллюстрирован разработанный нами алгоритм работы модуля подготовки G-кода из STL модели [0].

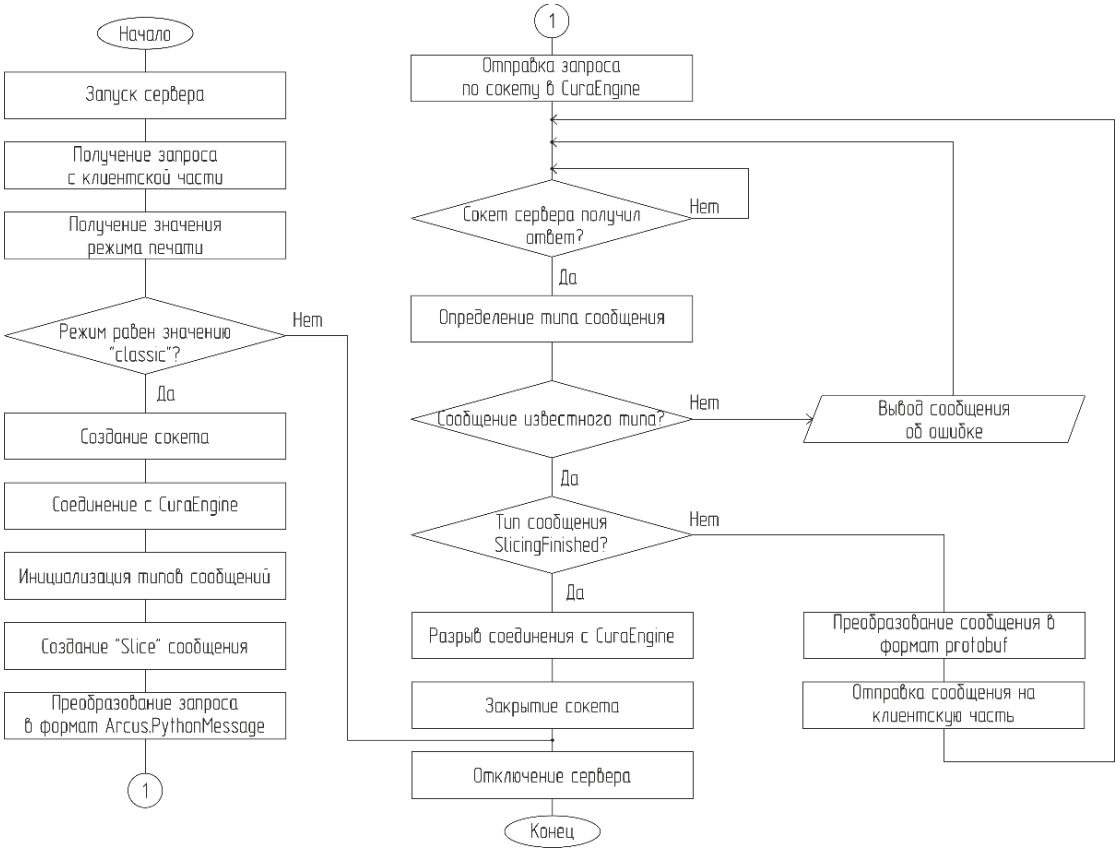

Рис. 2. Алгоритм обработки сервером запроса нарезки модели

Рассмотрим более подробно процесс общения с CuraEngine по сокету, для чего используется библиотека pyArcus [1761:a.i.5]. Она основана на libArcus для связи компонентов между программным обеспечением Ultimaker. Перед началом связи необходимо инициализировать используемый порт, IP адрес, описать используемые форматы сообщений, а также указать путь к самому CuraEngine и proto файлу, описывающего структуру существующих типов сообщений.

После инициализации необходимо проверить, запущен ли сокет в текущий момент, если он работает, то его нужно закрыть, а после создать "свежую" версию. Чтобы сокет понимал, как ему общаться с сокетом CuraEngine, в нем нужно зарегистрировать используемые типы сообщений. Сокет может быть запущен в двух форматах: либо как сервер, либо как клиент. Используя функцию listen, сокет запустится как сервер и будет прослушивать указанный ему порт, ожидая подключения клиента. Также необходимо добавить слушатель событий SocketListener, описанный на рис. 3, это позволит отслеживать состояние сокета и получаемые им сообщения.

```
123:  class _Listener(Arcus.SocketListener):
124:  def __init__(self, backend: Backend) -> None:
   1.       super().__init__()
   2.       self._backend = backend
            self._socket = backend._socket

   3.def stateChanged(self, state: Arcus.SocketState) -> None:
            print(f"[Socket] state changed: {state}", file=sys.stderr)

   4.def messageReceived(self) -> None:
   5.       message = self._socket.takeNextMessage()
   6.       if message.getTypeName() not in self._backend.type_messages:
   7.           print(f"[Socket] unkown message received: {message.getTypeName()}",
        file=sys.stderr)
   8.           return
   9.       print(f"[Socket] message received: {message}", file=sys.stderr)
            self._backend.getMessageHandlers[message.getTypeName()](message)

   10.  def error(self, error: Arcus.Error) -> None:
   11.          is_debug = error.getErrorCode() == Arcus.ErrorCode.Debug
            print(f"{'[Socket] debug' if is_debug else '[Socket] error'}:
        {error.getErrorMessage()}", file=(sys.stderr if is_debug else
        sys.stdout))
```

Рис. 3. Слушатель событий SocketListener

Сокет может иметь следующие состояния:

– Инициализация. Сокет не запущен, на этом этапе регистрируются сообщения;

– Подключение. Сокет был запущен как клиент, пытается подключиться к сокету-серверу;

– Открытие. Сокет был запущен как сервер, запускает внутренние функции;

– Прослушивание. Сокет ожидает подключения клиента;

– Подключено. Сокеты подключены между собой, сообщения могут отправляться и приниматься;

– Закрытие. Сокет закрывает соединение, но перед этим отправляет или принимает все сообщения в очереди;

– Закрыто. Сокет успешно закрыл соединение;

– Ошибка. Произошла ошибка, и сокет был отключен.

Когда сокет переходит из одного состояния в другое, получает сообщение, или возникает ошибка, слушатель событий реагирует на это и оповещает. После добавления слушателя и запуска сокета в серверном режиме, необходимо аналогично запустить сокет CuraEngine в режиме клиента. Для этого по пути расположения движка отправляется команда "connect ip:port". Если команда выполнится успешно, то сокет CuraEngine будет пытаться подключиться к сокету-серверу по указанному порту.

Библиотека subprocess позволяет параллельно запускать приложения и передавать им аргументы. С помощью этой библиотеки необходимо запустить сам CuraEngine, подключая

вывод движка к потоку стандартного вывода и потоку стандартного вывода сообщений об ошибках. Теперь при передаче сообщения Slice, в котором хранится информация о печатаемой модели и настройках печати, по сокету-серверу в сокет-клиент, движок сможет обработать эти данные аналогично тому, если бы ему отправляла эти данные Cura. По полученным данным CuraEngine строит модель и нарезает ее на слои для последующей генерации G-кода для каждого слоя по параметрам печати [1761:a.i.6]. Итоговый код CuraEngine отправляет обратно на сокет-сервер, где ответные данные обрабатываются и создается G-код файл, готовый для печати.

Используя разработанную нами систему, время подготовки 3D модели, например, червячного вала будет занимать не более 10 секунд, в то время как без нее только на запуск Cura уйдет около 15 секунд, а STL файл еще нужно открыть, установить настройки печати, нарезать модель на слои и экспортировать G-код, что в сумме может занять более 10 минут даже у профессионального пользователя.

Следующим шагом планируется автоматически отправлять готовый G-код на свободный принтер, а также учитывать параметры печати, которые вводят пользователи на сайте. Так пользователи смогут выбирать материал печати, цвет материала, плотность заполнения, режим печати и несколько других параметров.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Особенности развития 3D печати и возникающие при этом проблемы / У. Ф. Сухих, Т. Э. Замалеев, Ф. Ф. Ахметсафина, М. Ф. Шаехов // Новые технологии и материалы легкой промышленности : матер. XVIII Всероссийской науч.-практ. конф. с элементами научной школы для студентов и молодых ученых, Казань, 16–20 мая 2022 года. – Казань: Казанский национальный исследовательский технологический университет, 2022. – С. 108–110. – EDN YEXZEH.

2. Модуль подготовки деталей к печати в программном комплексе «Виртуальный 3D-принтер» / Е. А. *Синицин*, О. И. *Бритова*, В. В. *Попов* [и др.] // Молодежь в науке : сб. докл. 19-й науч.-техн. конф., Саров, 09–11 ноября 2021 года. – Саров: ФГУП «Российский федеральный ядерный центр – Всероссийский научно-исследовательский институт экспериментальной физики», 2022. – С. 150–156. – DOI 10.53403/9785951505200_150. – EDN LFUSZA.

3. Cura. [Электронный ресурс] // UltiMaker Cura – программное обеспечение для 3D-печати. – Режим доступа: https://ultimaker.ru/pages/ultimaker-cura/ (дата обращения: 26.06.2023).

4. *Grechukhin*, *A. N.* Algorithm for Dividing a Volume Model into Curvilinear Layers for 3D Printing / A. N. Grechukhin, V. V. Kuts, P. S. Shcherbakov // Proceedings of the 7th International Conference on Industrial Engineering (ICIE 2021) : ICIE: International Conference on Industrial Engineering, Sochi, 17–21 мая 2021 года. Vol. 2. – Челябинск: Springer, 2022. – P. 279-285. – DOI 10.1007/978-3-030-85230-6_33. – EDN PFRXZD.

5. Package libArcus. [Электронный ресурс] // libArcus-Java. – Режим доступа: https://ocarthon.github.io/libArcus-Java/docs/ (дата обращения: 26.06.2023).

6. *Grechukhin*, A. N. Model of Filling the Internal Structure of Workpiece with Curved Layers for 3D Printing / A. N. Grechukhin, V. V. Kuts // Materials Research Proceedings, Temryuk, 06–10 сентября 2021 года. – Temryuk, 2022. – P. 317-322. – DOI 10.21741/9781644901755-56. – EDN VPXMYY.