

РЕАЛИЗАЦИЯ КЛИЕНТ-СЕРВЕРНОГО ПОДХОДА В РАМКАХ РЕИНЖИНИРИНГА СИСТЕМЫ DIVA

Потапенко В.А
Кафедра ЭВМ, ДонНТУ
eline@cs.dgtu.donetsk.ua

Abstract

Potapenko V. The client-server architecture for the reengineering simulation system DIVA. The client-server architecture is reviewed as the method for the reconstruction of sophisticated systems of simulation complex technological processes.

Введение

Стремительное развитие средств вычислительной техники привело к тому, что многие трудоемкие разработки, ведущиеся годами, оказываются заложниками прогресса в области компьютерных технологий и не могут по ряду параметров быть приемлемыми в новых условиях. Особенно остро данный вопрос стоит для сложных систем, в частности, для систем моделирования. Одним из наиболее эффективных решений данной проблемы является реинжиниринг существующих систем. При этом максимально используется уже накопленный опыт и существующие разработки. Естественно, что реинжиниринг не всегда позволяет исправить все проблемы старых систем. Разработчики зачастую находятся в плену возможностей и ограничений уже существующей системы, что естественно находит отражение в новом продукте.

Каким образом выполняется реинжиниринг? Это в первую очередь зависит от тех недостатков, от которых желательно избавиться в новой системе. В последнее время наиболее актуальными являются следующие направления: перенос приложений в сетевую среду, оснащение системы новым или существенно улучшенным пользовательским интерфейсом, повышение быстродействия систем, добавление функциональности. В данной статье рассматривается пример реинжиниринга системы моделирования сложных динамических процессов DIVA [1].

Постановка задачи

Прототип моделирующей среды DIVA был разработан в 1987-90 гг. в институте системной динамики и управляющей техники Штуттгартского университета на базе проводившихся ранее в институте исследовательских работ в области разработки моделирующих систем технологической ориентации и моделирования отдельных технологических процессов и аппаратов. Система DIVA написана на языке программирования FORTRAN для операционной системы UNIX. DIVA не имеет графического пользовательского интерфейса, а работа с ней осуществляется в режиме командной строки.

DIVA позволяет:

- производить моделирование параметров процессов, которые не могут быть измерены непосредственно;

- определять причины отклонения текущих параметров процесса от заданных;
- обеспечить оптимальные условия реализации технологических процессов по критериям безопасности и производительности;
- использовать модели различных производственных процессов в учебном процессе при подготовке специалистов соответствующего профиля;
- подготавливать и контролировать производственный персонал.

Среди недостатков системы DIVA следует отметить следующие:

- графический пользовательский интерфейс отсутствует как таковой, имеется возможность задавать параметры работы используя командную строку и редактируя файлы заданий;
- отсутствует возможность работать в сетевой среде;
- отсутствуют средства распределенного моделирования.

Основным направлением реинжиниринга системы DIVA является реализация клиент - серверной архитектуры. Данный подход обладает рядом существенных преимуществ, среди которых:

- разделение системы на серверную и клиентскую части, что позволяет реализовать принцип "разделяй и властвуй", т.е. становится возможным поднять функциональность системы на существенно новый уровень;
- клиентская часть может быть реализована на базе "тонкого клиента", в то время как серверная на кластере мощных ЭВМ;
- администрирование системы при данном подходе значительно упрощается.

Клиент-серверный подход для организации взаимодействия

Так как в системе DIVA отсутствуют какие-либо средства для сетевой работы, то для обеспечения сетевого взаимодействия была реализована надстройка в виде дополнительных сетевых компонент. На сегодняшний день одним из лучших средств для создания сетевых компонент является среда программирования Java. Язык Java изначально разрабатывался как средство облегчающее создание сетевых приложений. Кроме того, Java является платформо-независимым языком, что позволяет выполнять приложения на различных вычислительных системах без какого-либо изменения исходного кода. В связи с этим, использование данного языка является наиболее приемлемым для дополнения системы моделирования DIVA сетевыми возможностями. Система DIVA построена таким образом, что задание условий и способов моделирования происходит из командной строки посредством текстовых файлов, передаваемых в качестве параметров к исполняемому модулю. Результатом работы системы DIVA также являются текстовые файлы. Таким образом, имеется возможность создать некую программную оболочку, которая бы осуществляла передачу извне к DIVA исходных параметров в виде привычных для нее текстовых файлов и отдавала бы результат работы обратно. Система DIVA не является системой моделирования реального времени. Входные параметры для расчета могут быть сформированы в любой момент времени, и после отработки результат становится доступным пользователю. Данная особенность значительно упрощает создание клиент-серверной моделирующей среды. В результате, взаимодействие осуществляется следующим образом. Пользователь на клиентской части формирует задание для выполнения его в среде DIVA. После этого по сети, используя протокол TCP/IP, задание передается серверной части. В общем случае взаимодействие может быть

реализовано в сети любого типа, поддерживающей протокол TCP/IP (локальная сеть, Internet/Intranet). Серверная часть, получив данные от клиентской, формирует стандартные для DIVA текстовые файлы задания на моделирование и запускает ее. Отработав, DIVA формирует файлы результата. Эти файлы преобразуются в поток данных и отсылаются клиентской стороне. В итоге, пользователь получает результат вычислений в виде данных и графиков. Данное взаимодействие изображено на рисунке 1.

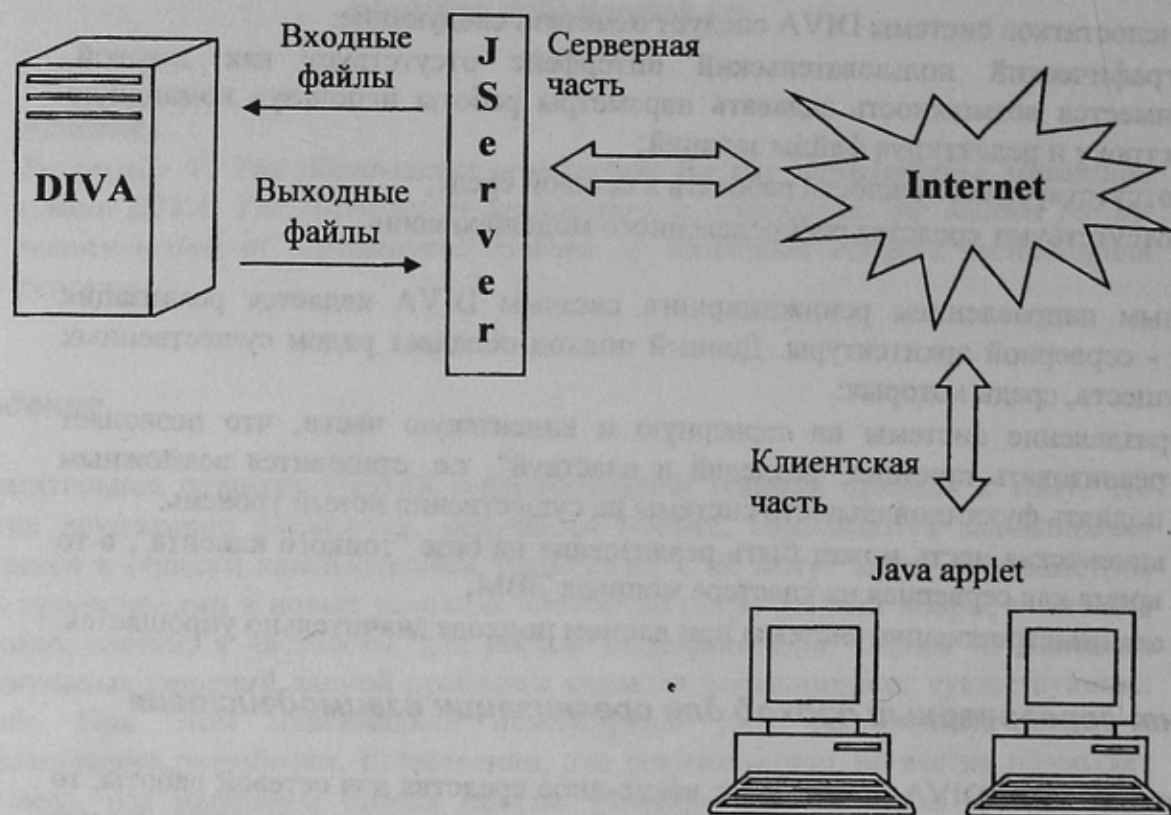


Рисунок 1. – Схема взаимодействия по технологии "клиент – сервер"

Для формирования пользовательского интерфейса на стороне клиента можно использовать широкий набор средств, включающий различные языки программирования. Данную преемственность можно объяснить тем, что серверная часть, выполненная на JAVA для сетевого взаимодействия, использует стандартизованные объекты класса «socket».

Протокол взаимодействия между клиентом и сервером

Организация распределенной моделирующей среды для системы «DIVA» в соответствии с клиент-серверной архитектурой предусматривает формирования набора правил и команд для взаимодействия между сетевыми компонентами, совокупность которых и составляет протокол программного уровня в стеке протокола TCP/IP.

Процесс взаимодействия между клиентской и серверной частями было предложено разделить на три этапа:

1. Установление соединения между клиентом и сервером.
2. Взаимодействие по установленному протоколу.
3. Разъединение соединения.

В свою очередь, этап взаимодействия предлагается разделить на две фазы:

1. Фаза диспетчеризации, иначе ожидание команд со стороны клиента.
2. Фаза выполнения команд клиента сервером.

Итак, первым этапом взаимодействия клиентской и серверной частей программного обеспечения является установление соединения. Это подразумевает, что клиент должен запросить разрешение на установление соединения, на что сервер, в свою очередь, должен своим ответом предоставить сетевой ресурс для обмена. Выражаясь терминами протокола TCP/IP, серверная часть слушает определенный порт, ожидая подключения клиента. После установления соединения этап взаимодействия переходит в фазу диспетчеризации. В этот момент времени сервер ожидает команд от клиента. При поступлении команды начинается фаза выполнения команд сервером. Вначале происходит анализ поступившей команды. В случае успешного распознавания команды непосредственное ее выполнения реализовано следующим образом:

1. Сервером осуществляется посылка подтверждения запроса на выполнение команды.
2. Непосредственное исполнение команды.
3. Подтверждение завершения исполнения команды.

Все команды представляют собой простые атомарные операции. Единственным исключением является команда передачи результатов моделирования клиенту. Реализация данной команды осуществляется в два этапа:

1. Осуществляется подтверждение завершения анализа результатов моделирования.
2. Ожидается подтверждение приема результатов моделирования от клиента.

Серверная часть программного обеспечения распределенной моделирующей среды для системы моделирования «DIVA» осуществляет выполнение следующего набора команд:

- получение файла данных от клиента;
- отправка результатов моделирования клиенту;
- подготовка файлов моделей для системы «DIVA»;
- запуск на выполнение моделирующей системы «DIVA»;
- разрыв соединения с клиентом.

Коротко рассмотрим функциональное назначение реализованного набора команд. Команда получения данных от клиента предназначена для передачи параметров модели серверу с целью последующего запуска системы «DIVA» на моделирование, используя полученные файлы.

Отправка результатов моделирования клиенту включает в себя также преобразование результатов моделирования системы «DIVA» к виду удобному для клиента.

Подготовка файлов моделей для системы «DIVA» осуществляется путем преобразования потока данных полученных от клиента в стандартные файлы заданий для «DIVA»;

Запуск на выполнение моделирующей системы «DIVA» осуществляется путем запроса к операционной системе на исполнение соответствующего модуля системы «DIVA».

По окончанию взаимодействия клиентом посылается запрос на выполнения команды, осуществляющей разрыв соединения.

Выводы

В настоящее время актуальность реинжиниринга существующих систем существенно возросла. Изменяются идеи и подходы к проектированию, появляются новые технологии, формируются новые требования, реализация которых зачастую не была первоначально заложена в проекте. Рассмотренный в данной статье подход к реинжинирингу системы DIVA может быть использован для придания новой функциональности многим существующим системам и представляется весьма перспективным.

Литература

1. Аноприенко А.Я., Кинле А., Святный С.Н., Осипова Т.Ф. Моделирование реактора синтеза уксусной кислоты на базе моделирующей среды DIVA / В кн. «Информатика, кибернетика и вычислительная техника (ИКВТ-97). Сборник научных трудов ДонГТУ.» Выпуск 1. Донецк, ДонГТУ, 1997, с. 16-21.
2. Забровский С.В., Мацак С.А., Потапенко В.А., Сисюкин В.С. Опыт использования сети Интернет в качестве распределенной среды моделирования. // Научные труды ДонГТУ. Серия «Проблемы моделирования и автоматизации проектирования динамических систем». Выпуск 29. – Д: ДГТУ, 2001. с 254-259.
3. Аноприенко А.Я., Ермоленко И.О., Потапенко В.А. Розробка компактних програмних засобів блочно-орієнтованого розподіленого моделювання динамічних систем. // Научные труды ДонГТУ. Серия «Проблемы моделирования и автоматизации проектирования динамических систем». Выпуск 10. – Д: ДГТУ, 1999. с 119-128.
4. Официальная страница языка Java. <http://www.java.sun.com>
5. Сисюкин В.С. “Методи та засоби реінженірінгу систем моделювання складних технологічних процесів на прикладі системи DIVA” // Выпускная работа магистра по специальности «Компьютерные системы и сети», 2001г.
6. Забровский С.В. “Методи і засоби графічної візуалізації в моделюванні складних систем” // Выпускная работа магистра по специальности «Компьютерные системы и сети», 2000г.
7. Nance, R.E. The Conical Methodology and the Evolution of Simulation Model Development," Annals of Operations Research, 53, Special Volume on Simulation and Modeling, O. Balci (Ed.), 1994, pp. 1-45.
8. John A. Miller, Rajesh S. Nair, Zhiwei Zhang and Hongwei Zhao. JSIM: A Java-based simulation and animation environment. In 30th Annual Simulation International (SCS), January Simulation Symposium, Atlanta, Georgia, April 1997.

Поступила в редакційну колегію 24.12.2001 р.