

ИСПОЛЬЗОВАНИЕ ТЕХНОЛОГИИ CORBA В РАСПРЕДЕЛЕННОМ МОДЕЛИРОВАНИИ СЛОЖНЫХ ТЕХНОЛОГИЧЕСКИХ СИСТЕМ

Аноприенко А.Я., Забровский С.В., Потапенко В.А.

Донецкий национальный технический университет, кафедра ЭВМ

Abstract

Anoprienko A.Y., Zabrovsky S.V., Potapenko V. A. Using CORBA technology in the distributed modeling of complex technologies systems. This article is about the methods and samples of using CORBA technology in distributed modeling and new ways of modeling with DIVA system.

В данной статье рассматриваются вопросы реинжиниринга системы моделирования DIVA[1], с применением технологии CORBA, с целью реализации распределенной среды моделирования сложных химических процессов. Проблема создания такой среды появилась вследствие необходимости обеспечения доступа к возможностям моделирующей системы DIVA большой группы пользователей с различными требованиями и возможностями аппаратного и программного обеспечения. Одним из путей решения данной проблемы является механизм распределенных объектов CORBA, позволяющий абстрагироваться от процессов разработки сетевого взаимодействия и мультиплатформенности.

CORBA – это архитектура брокеров распределенных объектных запросов[2]. Технология CORBA объединяет две важные тенденции в компьютерной индустрии – разработка объектно-ориентированного программного обеспечения и модель вычислений «клиент-сервер». CORBA – это не просто объектно-ориентированный вариант удаленного вызова процедуры RPC (Remote Procedure Call). Разработанная консорциумом OMG архитектура управления объектами OMA (Object Management Architecture) определяет несколько уровней абстрактности. Брокеры объектных запросов позволяют абстрагироваться от сложности разработок сетевого программного обеспечения. Службы CORBA обеспечивают классическую функциональность на уровне системы «в объектно-ориентированной манере». Средства CORBA предлагают стандартизованные методы решения проблем, характерных для определенной предметной области создания распределенных вычислительных систем. CORBA – это стандарт, позволяющий использовать ее на различных платформах, операционных системах и во взаимодействии с программами, написанными на различных языках программирования, достаточно только перекомпилировать некоторые исходные файлы и мы получим реализацию объекта для необходимой системы. Объект CORBA это абстрактный элемент, интерфейс взаимодействия с которым определяется с помощью языка определения интерфейсов IDL. Исполнитель это конкретный элемент языка программирования, обеспечивающий функциональность объекта CORBA. Запросом CORBA, как правило, является вызов клиентом метода удаленного объекта. Данный вызов для клиента предельно абстрактен, он сводится лишь к вызову метода локального программного объекта, интерфейс которого был описан на языке IDL, все остальное реализует брокер объектных запросов и созданные им заглушки и программы-заготовки.

Упрощенно схему работы CORBA–технологии можно представить следующим образом. Из рисунка 1 можно увидеть, что программный код клиента взаимодействует только с объектом-посредником, все остальное взаимодействие выполняет брокер объектных запросов.

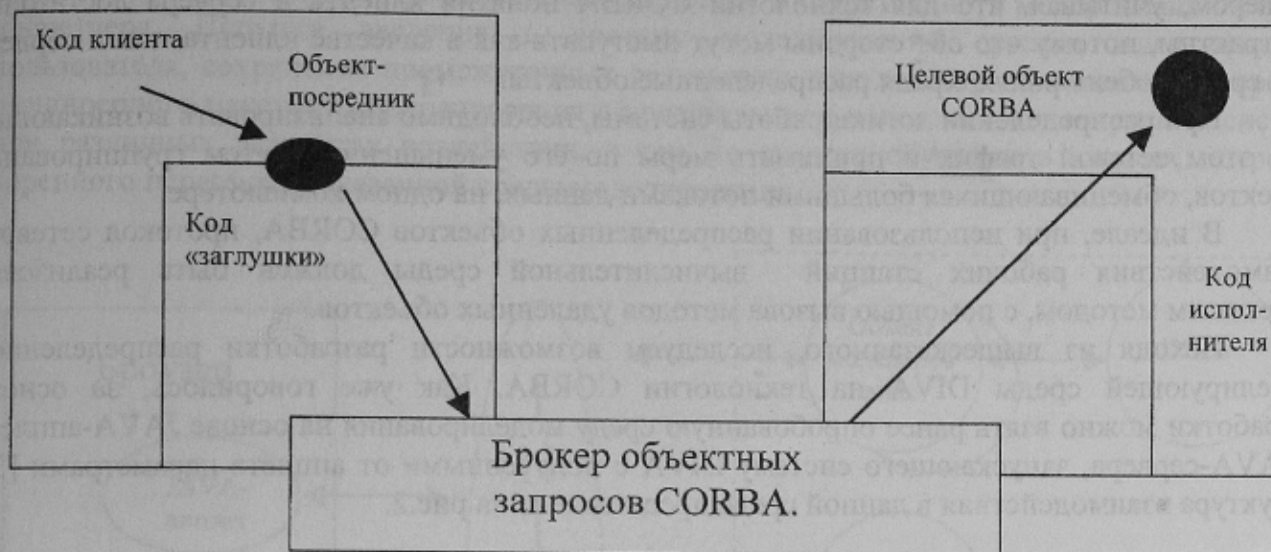


Рисунок 1- Схема вызова объекта CORBA

Для реализации объекта – посредника используется программная заглушка, предоставляющая для клиента интерфейс объекта – посредника, который был описан на языке IDL, и идентичен интерфейсу реального объекта, расположенного на сервере. Код заглушки и код программы – заготовки генерируется утилитами брокера объектных запросов из файла с описанием IDL – интерфейса. По сути дела после работы данной утилиты мы получаем исходные файлы заглушки и заготовки на необходимом языке программирования. Для их использования необходимо просто включить эти файлы в проекты клиента и сервера и откомпилировать. После этого в клиенте появился доступ к локальному объекту-посреднику, а в серверной части заготовка для описания исполнителей методов, весь код для удаленного взаимодействия был сгенерирован автоматически и прозрачен для программиста. Из рассмотренного можно сделать вывод, что программная заглушка, брокер объектных запросов и заготовка сервера объединяют процесс клиент и процесс сервер в единое целое, с точки зрения разработчика, выполняющееся на одной вычислительной машине, в одном процессе.

Рассмотренные особенности технологии CORBA позволяют успешно применять ее при реинжиниринге существующих систем. Поэтому было предложено использовать ее в процессе создания распределенной среды моделирования для системы DIVA. В качестве прототипа среды моделирования можно использовать уже не саму систему моделирования DIVA, а ее вариант, разработанный для работы в среде интернет на основе Java-апплета и Java-сервера [3]. Т.е. подвергнем реинжинирингу один из вариантов реализации распределенной системы моделирования.

Распределенная среда моделирования описанная в [3], обладая высокоорганизованным и удобным для использования интерфейсом с элементами и канонами когнитивной графики, а так же эффективно работающей серверной частью, имела существенный недостаток: это слабо реализованная поддержка распределенной обработки запросов на нескольких серверах с учетом нагрузочной способности и удаленности от клиента, что в свою очередь сказывалось на быстродействии в пакетных режимах работы или при подключении большого числа пользователей.

Перед разработкой системы на базе технологии CORBA необходимо провести множество исследований.

Прежде всего, необходимо разделить логику работы системы между клиентом и сервером, учитывая, что для технологии CORBA понятия клиента и сервера достаточно абстрактны, потому что обе стороны могут выступать как в качестве клиента, так и сервера или сразу в обеих ролях, храня распределенные объекты.

При распределении логики работы системы, необходимо анализировать возникающий при этом сетевой трафик и принимать меры по его уменьшению, путем группирования объектов, обменивающихся большими потоками данных, на одном компьютере.

В идеале, при использовании распределенных объектов CORBA, протокол сетевого взаимодействия рабочих станций вычислительной среды должен быть реализован объектным методом, с помощью вызова методов удаленных объектов.

Исходя из вышесказанного, исследуем возможности разработки распределенной моделирующей среды DIVA на технологии CORBA. Как уже говорилось, за основу разработки можно взять ранее опробованную среду моделирования на основе JAVA-апплета и JAVA-сервера, запускающего систему DIVA с полученными от апплета параметрами [3]. Структура взаимодействия в данной среде представлена на рис.2.

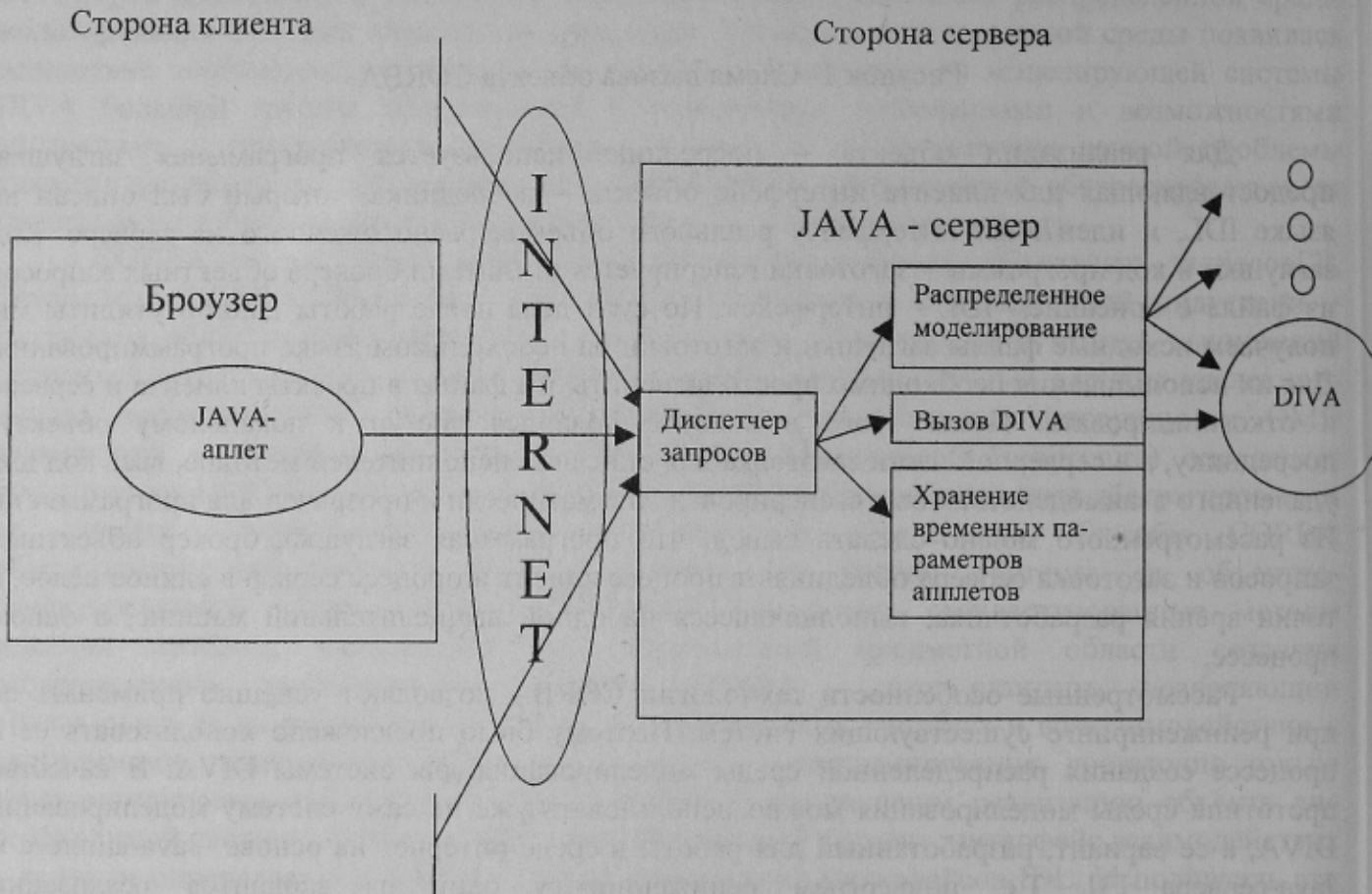


Рисунок 2 - Структура взаимодействия в распределенной моделирующей среде на базе JAVA-апплета и JAVA-сервера

В приведенной на рис.2 моделирующей среде взаимодействие между клиентом и сервером происходит по протоколу запрос - ответ, описанному в [3]. В процессе разработки выяснилось, что апплет вынужден хранить промежуточную информацию, чего нельзя сделать на стороне клиента, поэтому в серверную часть был введен модуль, отвечающий за это и диспетчер запросов, определяющий какого рода запрос пришел по протоколу. Однако подобное решение проблемы не всегда удобно, так как при дальнейшей модернизации системы могут возникать определенные трудности.

Во первых, практически любое нововведение будет приводить к необходимости введения новых команд в протокол взаимодействия а следственно к модернизации службы диспетчера. Попытка введения в систему моделирования средств идентификации пользователя, сохранения промежуточных результатов для данного пользователя и средств расширенного пакетного моделирования с предсказанием поведения моделируемой системы при различных вариантах воздействия, а так же средств обучения в классах потребует коренного пересмотра указанной среды моделирования.

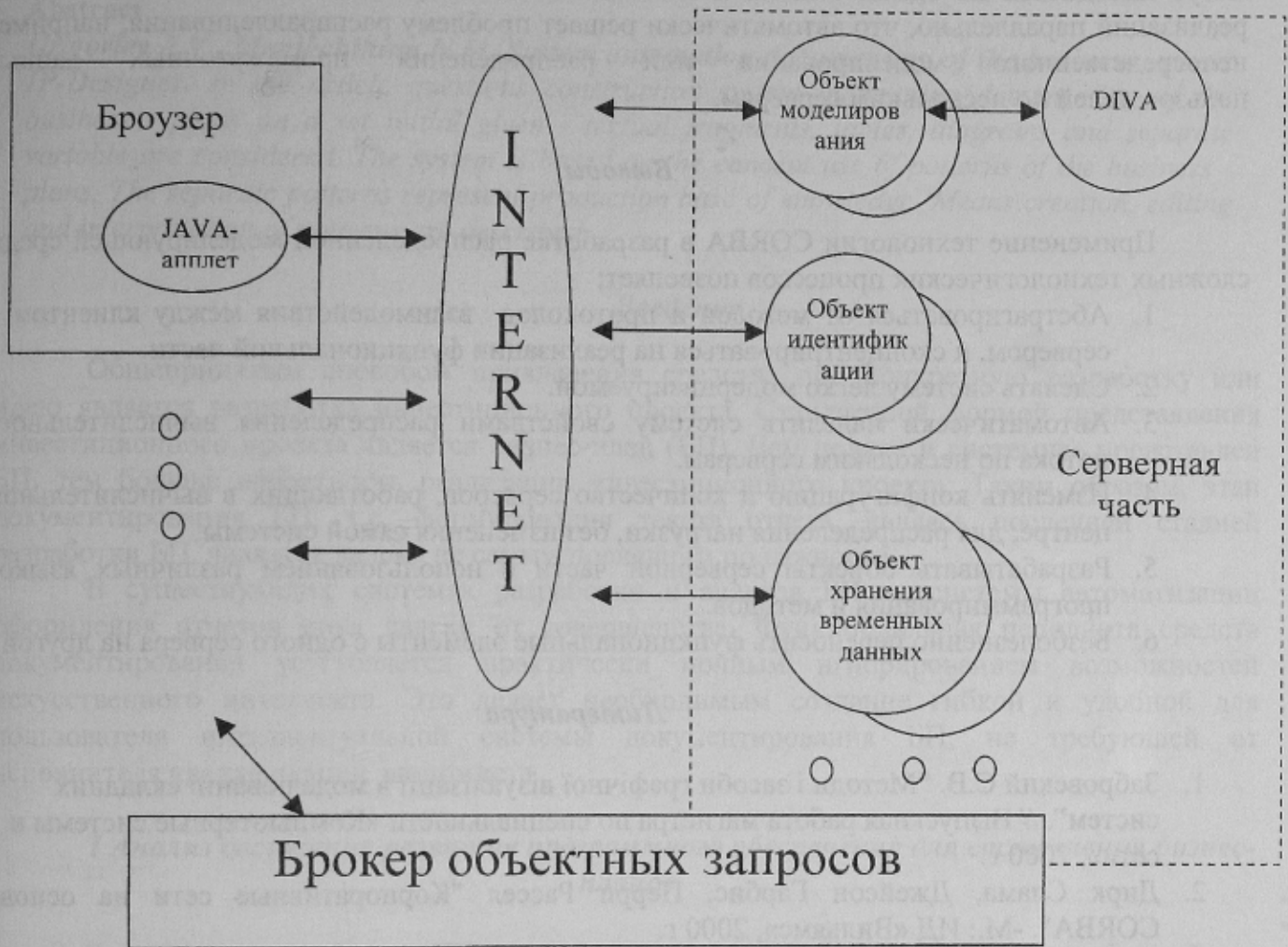


Рис.3 - Взаимодействие в распределенной моделирующей среде на базе технологии CORBA

Для более профессиональной среды моделирования потребуются более профессиональные подходы. Для повышения нагрузочной способности серверной части потребуется не только распределение DIVA по различным серверам и вызова наименее загруженного на моделирование, а и распределение обслуживающей части, отвечающей за все кроме непосредственного моделирования, по различным машинам.

Описанные выше проблемы и задачи можно с успехом решить, используя технологию распределенных объектов CORBA. Структура взаимодействия в распределенной моделирующей среде на базе технологии CORBA представлена на рис. 3.

Из рис.3 видно, что серверная часть моделирующей среды реализуется в виде множества объектов, каждый из которых отвечает за определенную функцию всей системы. Данное построение системы позволяет минимальными затратами модернизировать ее, для

добавления новой функции достаточно реализовать объект, описывающий данную функцию и указать апплету на существование нового объекта с определенным интерфейсом. Так же, при использовании CORBA все вопросы, связанные с вызовом, нахождением объектов и распределением нагрузки между серверами, на которых находятся объекты берет на себя брокер объектных запросов. Т.е. программисту нет необходимости в реализации протокола взаимодействия, описывается только функциональная часть объекта и методы его использования. Распределенные объекты CORBA, выделенные на рис.3 как серверная часть, могут находиться на одном или на нескольких компьютерах, а так же иметь несколько реализаций параллельно, что автоматически решает проблему распараллеливания, например непосредственного моделирования или распределения промежуточных данных пользователей по нескольким серверам.

Выводы

Применение технологии CORBA в разработке распределенной моделирующей среды сложных технологических процессов позволяет:

1. Абстрагироваться от методов и протоколов взаимодействия между клиентом и сервером, и сконцентрироваться на реализации функциональной части.
2. Сделать систему легко модернизируемой.
3. Автоматически наделить систему свойствами распределения вычислительного потока по нескольким серверам.
4. Изменять конфигурацию и количество серверов, работающих в вычислительном центре, для распределения нагрузки, без изменения самой системы.
5. Разрабатывать объекты серверной части с использованием различных языков программирования и методов.
6. Безболезненно переносить функциональные элементы с одного сервера на другой.

Литература

1. Забровский С.В. "Методи і засоби графічної візуалізації в моделюванні складних систем". // Выпускная работа магистра по специальности «Компьютерные системы и сети», 2000 г.
2. Дирк Слама, Джейсон Гарбис, Перри Рассел "Корпоративные сети на основе CORBA". -М.: ИД «Вильямс», 2000 г.
3. 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.
4. 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», 2001 г.
5. 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.