

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

УДК 004.9

С.Е. Попов

СИСТЕМА РАСПРЕДЕЛЕННЫХ ВЫЧИСЛЕНИЙ ДЛЯ ЭНТРОПИЙНОГО АНАЛИЗА ФУНКЦИОНАЛЬНЫХ ПОКАЗАТЕЛЕЙ УГОЛЬНЫХ ПРЕДПРИЯТИЙ

Введение

Технологии распределенных объектов в течение последнего десятилетия достигли высокого уровня развития. В связи с этим в различных областях сферы деятельности человека появляются программные решения, ориентированные на технологии такого уровня. Большой рынок занимают распределенные комплексы, предназначенные для управления банковскими системами (обслуживание банкоматов, системы перевода денежных средств и так далее), системы, работающие в сфере интерактивного развлечения (компания Hong Kong Telecom Interactive Multimedia Services, предоставляющая службу «видео по требованию», которая позволяет брать напрокат фильмы в электронном виде).

Появились распределенные системы для управления данными о продукции, планирования ресурсов предприятия и компьютерного планирования проектов (компания «Боинг», обслуживающая свои базы данных с конструкторской документацией при помощи таких систем) [1]. В то же время, несмотря на столь бурное развитие, существует ничтожно мало систем распределенных объектов, относящихся к горнодобывающей промышленности и позволяющих, в удаленном режиме, решать задачи управления геосистемами горного производства, а также вести геомеханический и газодинамический мониторинг.

Постановка задачи

В связи с актуальностью вышеописанной проблемы, была поставлена задача: на основе конкретного алгоритма разработать программный комплекс, реализующий спецификации распределенной системы, и на его примере создать методологическую базу построения систем распределенных объектов для решения широкого круга задач горного производства.

Разработка системы

Прикладной основой для построения программного комплекса послужила математическая модель, описанная в работе [2].

В качестве базовой была выбрана архитектура Microsoft .NET Remoting. Данная спецификация является последней разработкой Microsoft в сфере инструментальных средств для построения распределенных приложений. Она предоставляет набор классов для доступа к удаленным объектам на основе маршрутизации через границы доменов или контекстов объектов и заменяет собой устаревшую архитектуру DCOM.

Процесс разработки системы распределенных вычислений был разделен на несколько этапов.

На первом этапе была сделана попытка, в рамках выбранной архитектуры, проанализировать прикладной аспект задачи, то есть математическую модель, и построить программную или объектную модель задачи, а именно, выделить объекты следующих типов:

- объекты (классы), отвечающие за взаимодействие с базой данных, (т.к., выше представленный алгоритм оперирует с данными об угольных предприятиях и, следовательно, целесообразно хранить их на отдельном хосте с СУБД), причем, данный тип объектов должен содержать набор методов, которые выполняют действия только с источником данных на уровне SQL-инструкций и возвращают отсоединенные, необработанные, наборы данных (Data Sets) – Data Access Layer;

- объекты, отвечающие за «математику» алгоритма, т.е. расчет, представленных в нем формул, а также объекты, инкапсулирующие логику работы с отсоединенными наборами данных, полученных от классов Data Access Layer – Business Rules Layer;

- объекты, изолирующие бизнес-логику от изменений со стороны клиентской части и расположенные, в данном случае, между уровнем Business Rules и WinUI Layer (т.е. объекты, предоставляющие открытые классы и интерфейсы для клиентской части программного комплекса). Такой тип объектов называется Business Facade Objects - Business Facade Layer;

- объекты, обеспечивающие системные службы и функциональность системной инфраструктуры (в конкретном случае, это служба Windows, работающая как серверный домен приложения) - System Frameworks Layer;

- объекты клиентской части

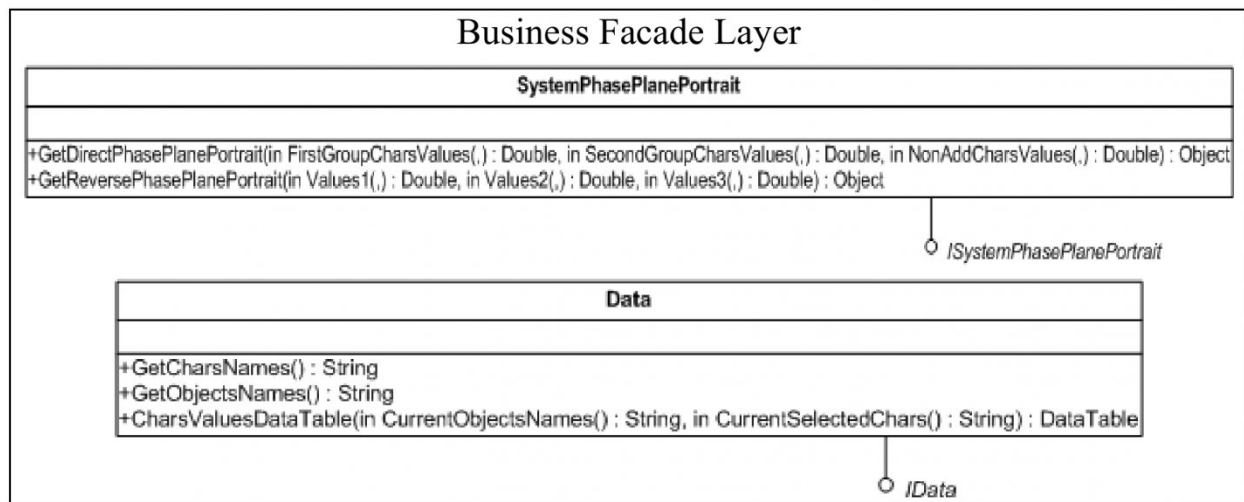


Рис. 1. Диаграмма классов Business Facade Layer

распределенного комплекса – WinUI Layer.

На втором этапе, с учетом вышеизложенных требований, была построена UML-модель

объектов, составляющих проект. Здесь, для наглядности, приведен только диаграммы классов серверной стороны.

Стоит заметить, что уро-

вень Business Facade содержит интерфейсы и соответствующие им классы, по структуре сходные с классами уровня Business Rules, за тем исключением, что

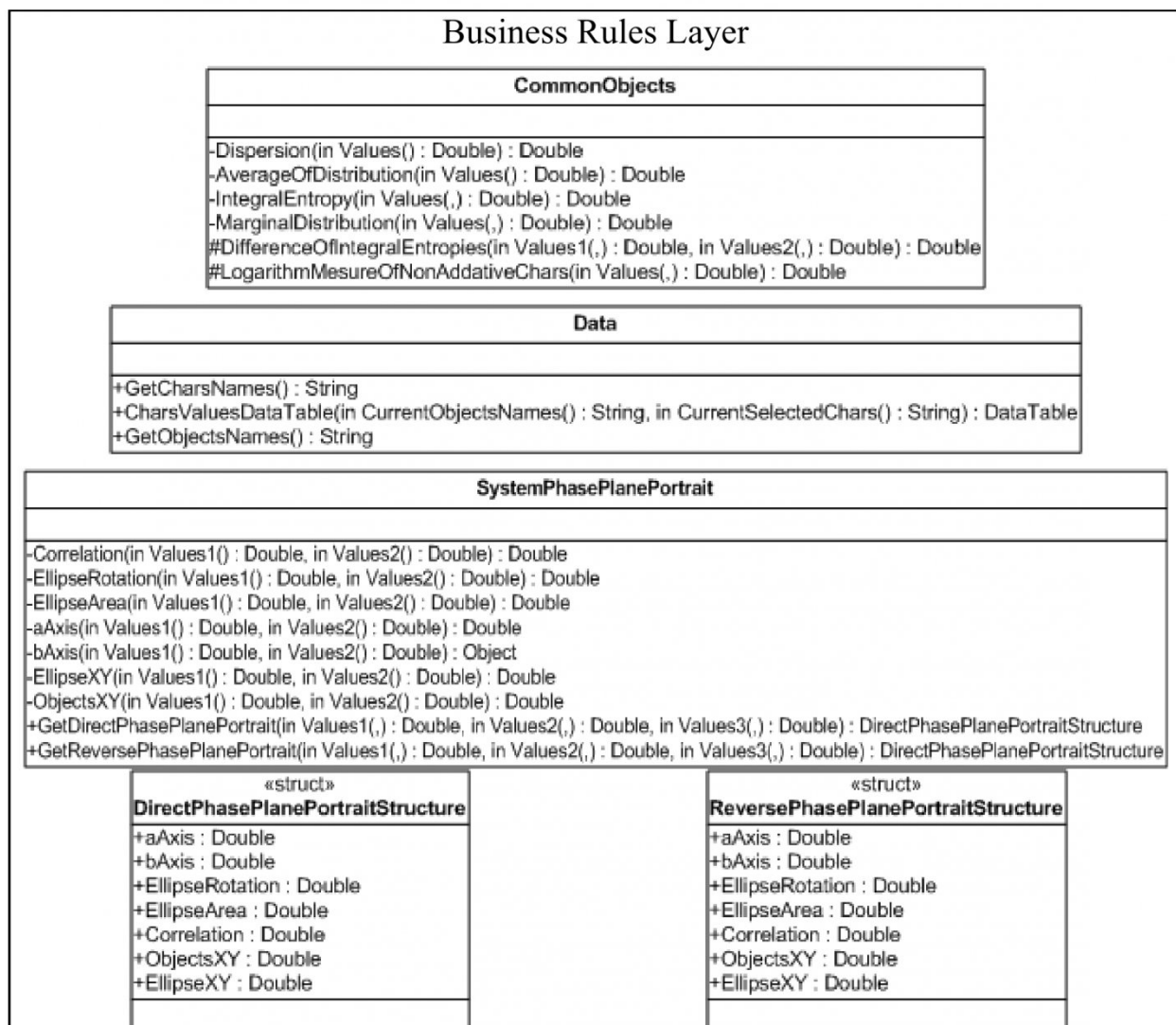


Рис. 2. Диаграмма классов Business Rules Layer

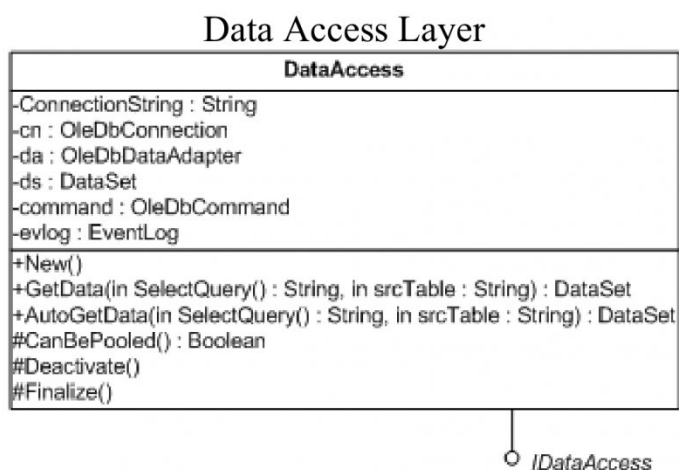


Рис. 3. Диаграмма классов Data Access Layer

программного каркаса проекта. На основе построенных UML-моделей были созданы классы соответствующих уровней (Data Access Layer, Business Rules Layer, Business Facade Layer, System Frameworks Layer, WinUI Layer). Генерация производилась в формате языка VB.NET. В качестве серверного домена приложения был выбран Windows Services-компонент, который при запуске открывает определенный порт (например – 4000 порт) и прослушивает его на предмет запросов к дистанцируемым объектам от удален-

Имя	Описание	Состояние	Тип запуска	Вход от имени
Alerter	Notifies se...		Вручную	Локальная сл...
Application Layer Gateway Service	Provides s...		Вручную	Локальная сл...
Application Management	Provides s...		Вручную	Локальная сис...
ASP.NET State Service	Provides s...		Вручную	.\ASPNET
Automatic Updates	Enables th...	Работает	Авто	Локальная сис...
Background Intelligent Transfer Service	Uses idle ...	Работает	Авто	Локальная сис...
ClipBook	Enables Cl...		Вручную	Локальная сис...
COM+ Event System	Supports ...	Работает	Вручную	Локальная сис...
COM+ System Application	Manages t...	Работает	Вручную	Локальная сис...
Computer Browser	Maintains ...	Работает	Авто	Локальная сис...
Cryptographic Services	Provides t...	Работает	Авто	Локальная сис...
DHCP Client	Manages ...	Работает	Авто	Локальная сис...
Distributed Link Tracking Client	Maintains l...	Работает	Авто	Локальная сис...
Distributed Transaction Coordinator	Coordinat...		Вручную	Сетевая служба
DNS Client	Resolves ...	Работает	Авто	Сетевая служба
EntropyAnalysis.PhasePlanePortraitWS		Работает	Авто	Локальная сис...
Error Reporting Service	Allows err...	Работает	Авто	Локальная сис...
Event Log	Enables e...	Работает	Авто	Локальная сис...

Рис. 4 Работаящий серверный домен приложения в виде Windows-сервиса

Активные подключения				
Имя	Локальный адрес	Внешний адрес	Состояние	PID
TCP	troll:http	troll:0	LISTENING	1192
TCP	troll:ermap	troll:0	LISTENING	780
TCP	troll:https	troll:0	LISTENING	1192
TCP	troll:microsoft-ds	troll:0	LISTENING	4
TCP	troll:1025	troll:0	LISTENING	804
TCP	troll:1027	troll:0	LISTENING	1192
TCP	troll:1032	troll:0	LISTENING	4
TCP	troll:1096	troll:0	LISTENING	3264
TCP	troll:1107	troll:0	LISTENING	3264
TCP	troll:4000	troll:0	LISTENING	1304
TCP	troll:5000	troll:0	LISTENING	908
TCP	troll:netbios-ssn	troll:0	LISTENING	4

Рис.5. Серверный домен приложения открыт на 4000 порту

в них инкапсулированы только те методы, которые имеют модификатор public на уровне Business Rules. Следовательно, включив вызовы этих методов в функции уровня Business

Facade, можно сформировать ограничения (изоляция) между объектами слоев Business Facade и Business Rules.

Следующим шагом в разработке системы стала генерация

ных клиентов (рис.3, 4).

На данную службу возлагаются функции по настройке инфраструктуры .NET Remoting. А именно, при запуске данного сервиса объявляется


```

<configuration>
  <system.runtime.remoting>
    <application name="EntropyAnalysis.PhasePlanePortraitService">
      <Service>
        <wellknown mode="Singleton"
          type="EntropyAnalysis.PhasePlanePortraitBF.SystemPhasePlanePortrait,
            EntropyAnalysis.PhasePlanePortraitBF"
          objectUri="SystemPhasePlanePortraitURI" />
        <wellknown mode="Singleton"
          type="EntropyAnalysis.PhasePlanePortraitBF.Data,
            EntropyAnalysis.PhasePlanePortraitBF"
          ObjectUri="DataURI" />
      </Service>
    <channels>
      <channel ref="tcp" port="4000" />
    </channels>
    </application>
  </System.runtime.remoting>
</Configuration>

```

Рис. 6. Файл PhasePlanePortraitBFServer.config в формате XML

```

<configuration>
  <system.runtime.remoting>
    <application name="EntropyAnalysis">
      <client>
        <wellknown
          type="EntropyAnalysis.PhasePlanePortraitBF.SystemPhasePlanePortrait,
            EntropyAnalysis.PhasePlanePortraitBF"
          url="tcp://remote.kemsc.ru:4000/SystemPhasePlanePortraitURI" />
        <wellknown
          type="EntropyAnalysis.PhasePlanePortraitBF.Data,
            EntropyAnalysis.PhasePlanePortraitBF"
          url="tcp://remote.kemsc.ru:4000/DataURI" />
        </client>
      <channels>
        <channel ref="tcp" port="0" />
      </Channels>
    </application>
  </System.runtime.remoting>
</Configuration>

```

Рис. 7. Файл PhasePlanePortraitBFClient.config в формате XML

имя домена приложения, в нем указывается вид и режим активизации, передаваемых по ссылке объектов, а также, сами эти объекты, их URI (Uniform Resource Identifier); указывается тип канала (tcp и/или http) и порт. Настройка может производиться как программно, так и при помощи конфигурационного файла. В нашем случае, конфигурационный файл представляет собой документ в XML формате (рис. 6).

Вызов этого файла можно добавить в процедуру запуска службы Windows при помощи оператора:
RemotingConfiguration.Configure ("имя_файла.config") [3].

Необходимо учесть, что при разработке клиентской части программного комплекса инфраструктура .NET Remoting должна быть сконфигурирована с учетом настроек серверной части, т.е. содержать либо программную настройку, либо вызов конфигурационного файла (рис. 7).

Тип маршалируемых объектов и тип порта должен совпадать с настройками, указанными на стороне сервера, имя приложения, вызывающее удаленные объекты, должно быть изменено на имя сборки клиентской части программы, в качестве номера порта выбрано значение 0, что позволяет инфраструктуре .NET

Remoting устанавливать любой свободный порт на стороне клиента [3].

Последним этапом в разработке комплекса стало его развертывание, как на стороне сервера, так и на стороне клиента.

В отличие от технологии Microsoft DCOM, .NET Remoting не требует регистрации и установки тех или иных удаленных компонентов, все, что необходимо сделать это переписать файлы проекта в любую папку и запустить программу на выполнение, со стороны клиента, и службу Windows – со стороны сервера.

Инфраструктура .NET Remoting сама настроит все необ-

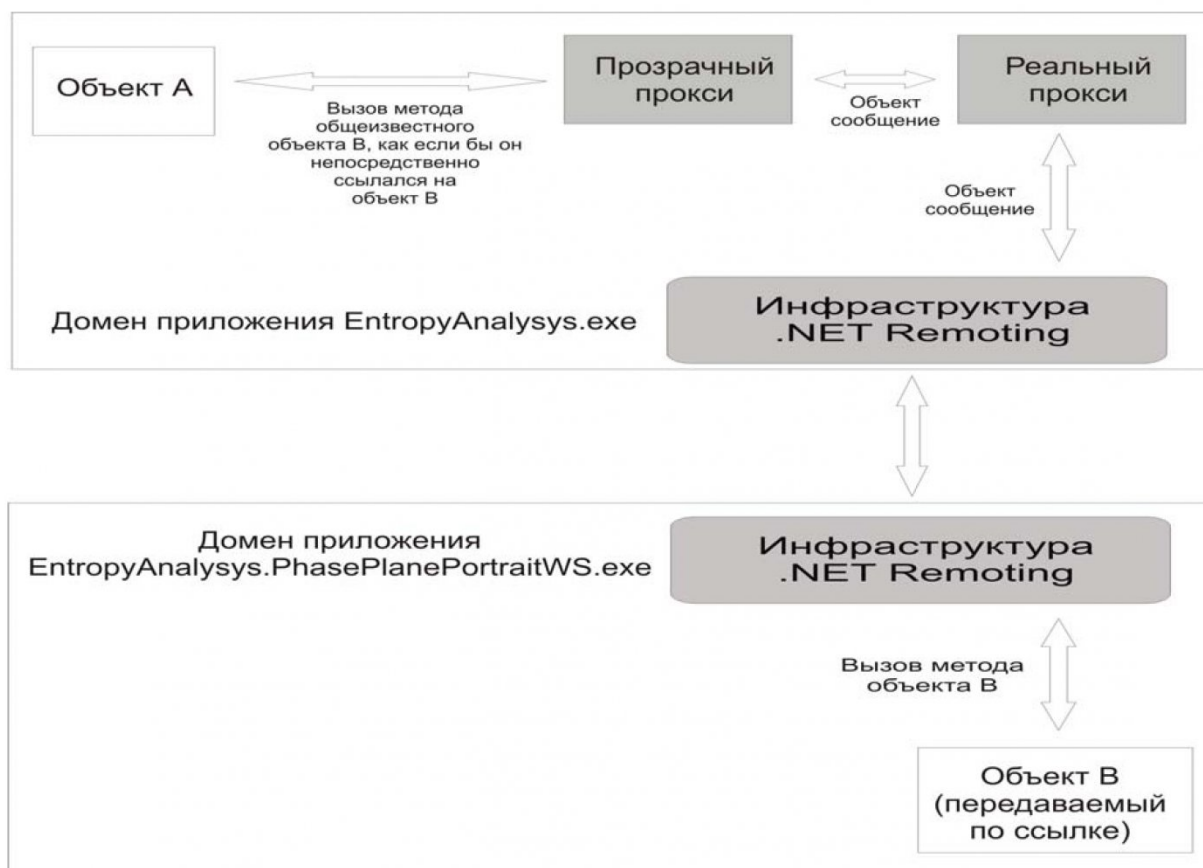


Рис. 8. Схема взаимодействия клиентов с удаленными объектами

ходимые компоненты с учетом конфигурационных файлов. Единственным условием, накладываемым на операционную среду, в которой будет установлен комплекс, является: установленная среда .NET Framework версии 1.0 и выше.

В общем виде, схема работы комплекса представлена на рис.8.

Выводы

В результате проделанной работы был разработан программный комплекс, позво-

ляющий оценивать деятельность угольных предприятий в удаленном режиме, а именно графически моделировать задачи двух типов: определение рейтинга объекта (шахты) по совокупности показателей (прямая задача); определение рейтингов показателей в системе объекта (шахты), т.е. строить фазовые портреты системы уникальных объектов. В качестве тестового примера были использованы экономические показатели 2001 года, такие как

Добыча, Производительность, Себестоимость, Численность персонала, которые рассматривались на 77 объектах (шахтах). Тем самым, была сформирована общая методология построения систем распределенных вычислений, которая позволит в уже ближайшее время решать широкий класс задач управления геосистемами горного производства, а также, создать новый класс систем геомеханического и газодинамического мониторинга.

СПИСОК ЛИТЕРАТУРЫ

1. Эммерих В. Конструирование распределенных объектов. Методы и средства программирования интероперабельных объектов в архитектурах OMG/CORBA, Microsoft/COM и Java/RMI. / Пер. с англ. – М.: Мир, 2002. 510 с.
2. Логов А.Б., Кочетков В.Н., Рожков А.А. Энтропийный подход к моделированию процесса реструктуризации угольной отрасли. Институт Угля и Углекислоты СО РАН, Кемерово-М.: 2001, 324 с.
3. Маклин С., Нафтел Дж., Уильямс К. Microsoft .NET Remoting/Пер. с англ. – М.: Торгово-издательский дом «Русская редакция», 2003. 384 с.

□ Автор статьи:

Попов
Семен Евгеньевич
- аспирант Института Угля
и Углекислоты СО РАН