

А.Р.Ешкеев, Р.М.Оспанов, О.И.Ульбрихт

Δ -йонсондық теориялардың экзистенциалды формулалардың торы

Мақалада жаңа Δ -йонсондық теориялардың кейбір модельді-теориялық және экзистенциалдық формулалардың торына қатысты йонсондық теориялардың орталық толығының кейбір қасиеттері қарастырылды. Осы ұғымдардың арасындағы модельдер теориясында белгілі сұрақтарға сай байланыстар дәлелденді. Δ -йонсондық теориялардың кемелдегі және Линденбаум-Тарский алгебраларымен байланысты сұрақтар қарастырылды.

A.R.Eshkeev, R.M.Ospanov, O.I.Ulbriht

The lattices of existential formulas of Δ -jonsson's theories

We are considering some model-theoretical properties of Δ -Jonsson's theory regarding the lattice of existential formulas of this theory. With this subject we are proved theorems about relations between these items. It is turns out that properties of perfectness of Jonsson's theories had description in terms of different kinds of completeness of the lattice of existential formulas of this theory. Also we considered links with famous question of A.D.Taimanov on Boolean algebras.

УДК 615.036.2

С.К.Жумагулова¹, Б.М.Саданова²

¹Карагандинский государственный университет им. Е.А.Букетова;

²Карагандинский государственный технический университет (E-mail: - saulesha_81@mail.ru)

Исследование и разработка средств поиска в локальных коллекциях документов

В статье рассмотрен набор компонентов, который позволяет организовать поиск документов в интранет с использованием системы поиска на локальном компьютере Google Desktop Search. Приведено описание предлагаемого способа поиска информации в локальных сетях, а также принципиальная схема функционирования предлагаемого решения. Изучен алгоритм реализации данной системы в качестве Windows-сервисов. Охарактеризованы компоненты сервера. Проанализированы планы расширения функциональности приложения несколькими способами.

Ключевые слова: компонент хранилища, метаданные, пространство имен System.Collections, .NET Remoting, компонент сервера.

Трудно переоценить значение использования компьютера в современных предприятиях и организациях. Практически все процессы в них связаны с прямым или косвенным использованием компьютеров. Объемы хранилищ документов организаций исчисляются гигабайтами, а в некоторых случаях и терабайтами. При таких объемах поиск нужного документа (особенно в слабоструктурированных хранилищах) становится достаточно сложной проблемой. Он усугубляется тем, что часто в организации имеется несколько хранилищ документов, расположенных на разных серверах, и ручной поиск документа в таком случае становится практически невозможным.

Поэтому создание эффективных поисковых машин в электронных хранилищах организаций является актуальной задачей. Необходимо создать набор компонентов для поиска информации в локальных сетях с распределенными или сосредоточенными на одном компьютере хранилищами документов с использованием существующих (желательно бесплатных) программных средств.

Основным результатом выполненной работы является разработанный с использованием технологии .NET набор компонентов, который позволяет организовать поиск документов в интранет с использованием системы поиска на локальном компьютере Google Desktop Search [1].

Выбранный подход, а именно «разработка интерфейса (оболочки — *wrapper*) для «обертывания» существующих программ или существующих компонентов сторонних разработчиков» может быть обоснован относительно малой трудоемкостью реализации и большой эффективностью.

Описание предлагаемого решения:

- 1) берется существующая поисковая система (Search engine), которая работает эффективно, но не удовлетворяет каким-то потребностям;
- 2) разрабатывается компонент-оболочка, которая, взаимодействуя с поисковой системой, решает поставленную задачу;
- 3) создается компонент, который организует взаимодействие с разработанной оболочкой.

Принципиальная схема предложенного решения представлена на рисунке 1.

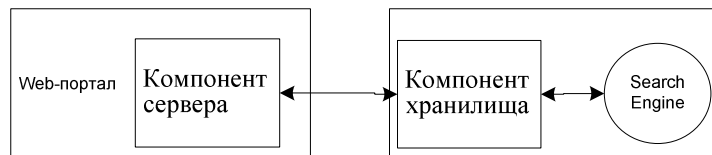


Рисунок 1. Принципиальная схема функционирования предлагаемого решения

Это решение предполагает создание системы, включающей две основные части:

1) компонент, устанавливающийся на компьютер, к которому будут адресованы запросы конечных пользователей (далее — *компонент сервера*). Именно он организует взаимодействие с разработанной оболочкой для поисковой системы;

2) множества компонентов, устанавливающих на компьютеры, где находятся хранилища документов (далее — *компонент хранилища*). Именно он взаимодействует с поисковым движком для получения результатов поиска.

Основные части системы реализованы в качестве Windows-сервисов [2].

Компонент сервера может быть подключен к программе, которая нуждается в предоставляемой ей функциональности. Это может быть Web-сервер, некоторое приложение, которое предоставляет доступ к компоненту на основе Windows-форм и т.п. Получая запрос от подключившего его приложения, компонент рассылает его всем или нескольким зарегистрированным в нем компонентам хранилищ (в зависимости от того, указан ли в запросе набор хранилищ, интересующий пользователя). Затем он собирает ответы, полученные от компонентов хранилищ и передает их вызывавшему приложению.

Компонент хранилища несет в себе больше функциональности. Получая запрос от компонента сервера, он передает его GDS, затем разбирает его и каждый результат сохраняет как объект класса *Google Answer*. В результате в имеющемся массиве результатов выбираются те, доступ к которым разрешен для запрашивающего пользователя (папки, к которым разрешен доступ, задаются настройкой компонента). Эти выбранные результаты возвращаются компоненту сервера.

Принципиальная архитектура системы на основе компонентов изображена на рисунке 2.

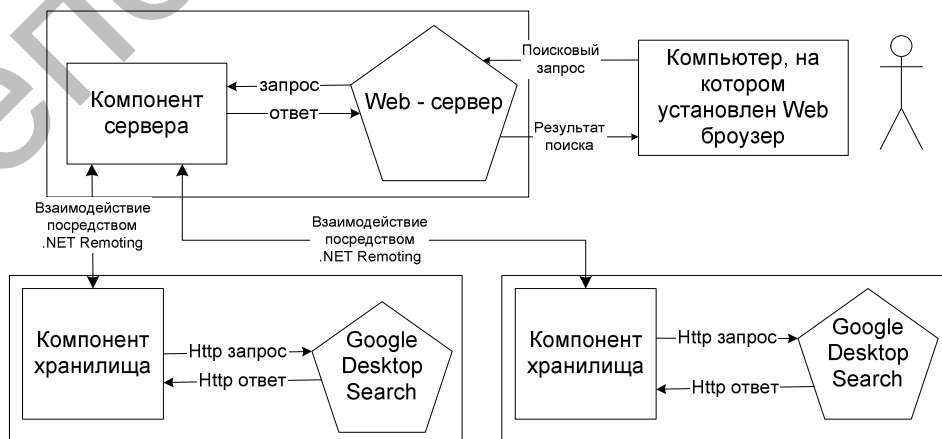


Рисунок 2. Архитектура поисковой системы

Принципиальным решением при построении системы является выбор системы взаимодействия между компонентом сервера и компонентами хранилищ. Рассмотрим два основных варианта — COM/DCOM и .NET Remoting. Значительная часть мощи и простоты использования объектной технологии обусловлена лежащими в ее основе системой типов и моделью объектов. В DCOM действуют те же ограничения, которые присущи системе типов и объектно-ориентированным возможностям COM. COM не поддерживает наследование реализации, за исключением конструкций, подобных агрегированию и включению. Средства обработки ошибок ограничены кодами возврата, так как COM не поддерживает исключения. Система типов COM также является непоследовательной и разрозненной. COM-системы на основе языка C++ используют описания типов в исходных текстах (IDL), тогда как Visual Basic и языки сценариев — двоичные описания типов (библиотеки типов). Ни IDL, ни библиотеки типов не могут считаться безусловным стандартом, так как каждый из этих вариантов поддерживает конструкции, не поддерживаемые другим вариантом. Наконец, COM не поддерживает целый ряд объектно-ориентированных конструкций, таких как модификаторы `static`, виртуальные функции и перегружаемые методы.

В противоположность DCOM .NET Remoting является мощной и простой в использовании технологией во многом благодаря общей системе типов (*common type system*, CTS) и общезыковой исполняющей среде (*common language runtime*, CLR). Информация о типах — метаданные — стандартизована и доступна. CTS определяет набор базовых типов, которые должны поддерживаться всеми .NET-совместимыми языками. Эти элементы так же надежны при удаленном взаимодействии, как и при взаимодействии между классами внутри одной программы. Более того, метаданные имеют унифицированный формат и хранятся вместе с определяющей их сборкой (*assembly*), поэтому удаленным объектам не требуются отдельные описания типов, как в DCOM и CORBA.

Благодаря возможности использовать всю мощь объектно-ориентированных средств .NET, .NET Remoting полностью поддерживает наследование реализации, свойства, а также статические, виртуальные и перегружаемые методы. CLR и CTS позволяют разработчикам использовать одну объектную систему как для локальных, так и для удаленных объектов и избегать проектных решений, в которых удаленное расположение объектов ограничивает свободу использования объектно-ориентированного подхода. Наконец, .NET полностью поддерживает распространение исключений между удаленными процессами, что значительно упрощает обработку ошибок для распределенных объектов по сравнению с кодами возврата DCOM [3].

Исходя из приведенных аргументов, для решения поставленной задачи был выбран более новый, более удобный .NET Remoting.

В наши дни существует несколько завоевавших успех на рынке технологий созданий распределенных приложений — CORBA, Java RMI, DCOM. Все они более или менее полно удовлетворяют запросы пользователей, хотя прежде, чем это стало так, каждая из этих технологий прошла немалый путь.

С появлением .NET возникла альтернатива этим технологиям — .NET Remoting. Одно из главных отличий Remoting от предшественников состоит в том, что эта технология создана для работы в управляемой среде. Там, где раньше речь шла о процессах и взаимодействии между ними, теперь говорится о контекстах и доменах приложений. Дело в том, что CLR контролирует типы в управляемом коде и проверяет, не производится ли обращений по недопустимым адресам. Это позволяет нескольким управляемым приложениям исполняться внутри одного процесса так же изолированно, как и в неуправляемой модели, где каждому из приложений отводится собственный процесс.

.NET Remoting — это технология управляемых удаленных вызовов между доменами приложений. Домены приложений — это единицы изоляции в CLR, которые создаются и работают внутри процесса. Код и объекты, исполняющиеся в одном домене приложения, не имеют непосредственного доступа к коду и объектам, исполняемым в другом домене. Именно это определяет сферу применимости Remoting.

На рисунке 3 приведена схема, иллюстрирующая работу .NET Remoting.

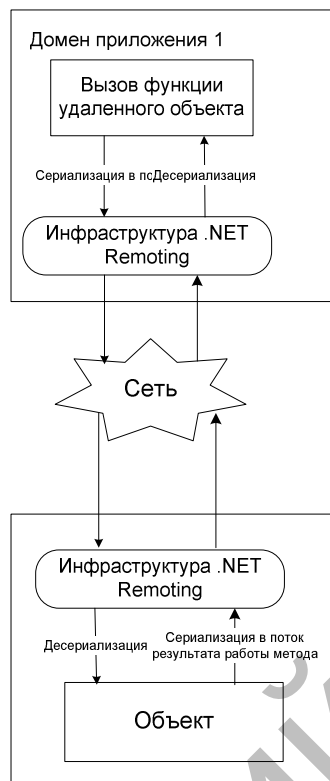


Рисунок 3. Принципиальная схема функционирования .NET Remoting

Программисту для работы с удаленными объектами теперь достаточно лишь зарегистрировать классы некоторым образом и затем можно не задумываться о том, что объекты этих классов — удаленные.

Описание компонента сервера.

Компонент сервера обладает следующей функциональностью:

I. Поисковая функциональность:

- 1) на основе запроса, полученного от использующего его приложения, формировать запрос к Google Desktop Search;
- 2) осуществлять взаимосвязь с зарегистрированными в нём компонентами хранилищ;
- 3) обладать способностью рассылать запрос и получать ответ от компонентов хранилищ;
- 4) обеспечивать поддержку разграничения поисковых прав в зависимости от принадлежности пользователя, осуществляющего поиск, к тем или иным поисковым группам;
- 5) обеспечить выборочную рассылку к некоторой группе поисковых хранилищ для поддержки функционирования групп хранилищ;
- 6) быть устойчивым к разрывам сети;
- 7) обеспечивать функциональность сервера вне зависимости от того, аутентифицирован ли какой-либо пользователь в системе;
- 8) содержать информацию о текущем состоянии поисковых хранилищ;
- 9) кэшировать результаты поиска.

II. Функциональность по настройке поисковых хранилищ.

- 1) обеспечивать возможность удаленной настройки поискового хранилища, то есть задания поисковых групп и разрешенных для них путей.

Использованные в компоненте пространства имен .NET Framework

- Пространство имен System.Collections.

Пространство *System.Collections* библиотеки классов .NET Framework содержит классы, служащие контейнерами для групп или наборов (collections) данных. Один из примеров — *ArrayList*, реализующий массивы переменной длины. Благодаря этому и другим типам, определенным в *System.Collections*, можно потратить больше времени на написание кода, специфичного для вашего приложения, и меньше — на утомительное написание инфраструктурного кода [4].

В реализованном наборе компонентов активно использовался класс *ArrayList* как контейнер для объектов класса *Google Answer*.

- Пространство имен *System.Runtime.Remoting*.

Пространство имен *System.Runtime.Remoting* обеспечивает возможность взаимодействия между удаленными классами через .NET Remoting.

- Пространство имен *Microsoft.Win32*.

Содержит классы двух типов: классы, обрабатывающие события, вызываемые операционной системой, и классы, работающие с системным реестром. Предназначено для извлечения части настроек из реестра, в том числе о месторасположении файла конфигурации.

Набор компонентов разрабатывается таким образом, чтобы его функциональность могла быть легко расширена. В дальнейшем планируется расширить функциональность приложения следующим образом:

- добавить возможность искать в некоторых выбранных каталогах (при условии что к ним у пользователя имеется доступ);
- расширить спектр информации, получаемой от компонентов хранилищ о каждом конкретном файле такими сведениями, как:
 - размер;
 - дата создания;
 - дата изменения;
 - создать возможность поиска и по этим параметрам;
- реализовать возможность работать с помощью .NET Remoting через HTTP, а не через TCP, что даст возможность использовать систему в тех организациях, где филиалы не соединены единой локальной сетью, а связь между локальными сетями филиалов осуществляется через Интернет и для безопасности защищена брандмауэрами.

Разработка систем поиска в локальных сетях является актуальной на сегодняшний день задачей. Скорость нахождения необходимой информации — фактор, в значительной мере сказывающийся на производительности труда сотрудников, а соответственно, и на производительности организации в целом. Показателем того, что такие системы необходимы, является наличие большого выбора подобных систем на рынке программного обеспечения [5].

Разработанное приложение имеет достаточно большие перспективы, так как основано на бесплатном движке поиска и в то же время предоставляет достаточно широкий спектр поисковых услуг.

Показателем того, что такие системы необходимы, является наличие большого выбора подобных систем на рынке программного обеспечения [5].

References

- 1 *Jef Prosz.* Programming for .NET / Russian edition. — Moscow, 2003.
- 2 *McLean S., Naftel J., Williams K.* Microsoft .NET Remoting / Russian edition. — Moscow, 2003.
- 3 *Gunnerson E.* Introduction in C# — St. Petersburg, 2001.
- 4 *Michael Dz.* Young Microsoft XML // Step by step. — Moscow, 2000.
- 5 *Richter G.* Programming on the Microsoft .NET Framework/Platform Russian edition. — Moscow, 2003.

С.К.Жумагулова, Б.М.Саданова

Күжаттарды локалды коллекцияларда іздеу құралдарды зерттеу және құру

Мақалада Google Desktop Search локалды компьютерде іздеу жүйесін пайдаланумен интранетте құжаттарды іздеуді ұйымдастыруға мүмкіндік беретін компоненттер жиынтығы қарастырылған. Ұсынылатын локалды желілерде ақпаратты іздеу тәсілі берілген. Бұл жүйені Windows-сервис арқылы іске асыру алгоритмі зерттелген. Сервер компоненттері анықталған. Қосымшаның жұмысын кеңейту мақсатымен әр түрлі тәсілдер жоспарлары келтірілген.

S.K.Gumagulova, B.M.Sadanova

Research and development of means of search in local collections of documents

In article the set of components which allows to organize search of documents in the intranet with use of system of search on the local Google Desktop Search computer is considered. The description of an offered way of information search is provided in local networks, and also the schematic diagram of functioning of the offered decision. The algorithm of realization of this system as Windows-services is considered. Server components are described. Plans of expansion of functionality of the appendix are provided in several ways.