

РАЗРАБОТКА РАСПРЕДЕЛЕННОЙ ГИС ИНЦ СО РАН*

И. В. БЫЧКОВ, Е. Л. КУХАРЕНКО

Институт динамики систем и теории управления СО РАН

Иркутск, Россия

e-mail: bychkov@icc.ru

The problems of integrating the information of various types and formats into a single system are discussed which would provide cartographic representation and access via corporate network as well as remote access through the Internet browsers and result in the development of an integrated GIS.

В институтах Иркутского научного центра (ИНЦ) СО РАН накоплена уникальная информация о природных ресурсах, экологической обстановке, экономике и т. д. региона, которая систематизирована и хранится в разнородных локальных базах данных с использованием различных СУБД, аппаратных платформ и прикладного программного обеспечения. Для проведения комплексных междисциплинарных исследований возникает необходимость интеграции разнотипной и разноформатной информации в единую систему, обеспечивающую картографическую форму представления и доступ как через корпоративную сеть, так и (удаленный) посредством браузеров Интернет. Таким образом, создание интегрированной ГИС предполагает применение архитектуры распределенных систем обработки информации. Данная архитектура позволяет оптимально использовать дорогостоящее аппаратное (сервер ГИС, сканер, плоттер) и программное (MapInfo, ARC/INFO и т. д.) обеспечение. Для реализации предлагаемого подхода разрабатываются связующие программные компоненты на Delphi v.3.0. В ГИС центре ИНЦ СО РАН имеющееся чисто лицензионное программное обеспечение базируется на платформе операционных систем Windows, поэтому при разработке архитектуры построения системы удаленного доступа к указанным ресурсам рассматривались только решения фирмы Microsoft. Развитие технологий обмена функциональными возможностями и данными между сетевыми приложениями, предлагаемые фирмой Microsoft, можно представить следующим образом:

DDE (dynamic data exchange) — динамический обмен данных;

OLE (object linking and embedding) — связывание и внедрение объектов;

COM(Common Object Model) — общая объектная модель, или модель компонентных объектов;

DCOM(Distributed COM) — общая модель распределенных объектов, по определению “удаляющая” реализацию интерфейса от точки исполнения;

*Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований, грант №98-07-90314.

© И. В. Бычков, Е. Л. Кухаренко, 1998.

ActiveX — специализированный набор средств, построенных на базе COM и предназначенных для решения задач, связанных с World Wide Web и корпоративными сетями (интрасетями).

В настоящее время технология DCOM является основой для построения системы, обеспечивающей объединение информационных ресурсов, оборудования, программных систем, взаимодействие между СУБД, ГИС, других приложений и возможность доступа к ним через World Wide Web.

Рассмотрим более подробно технологии, используемые при разработке распределенной ГИС ИНЦ СО РАН.

Обычно приложение состоит из одного монолитного двоичного файла, сгенерированного компилятором, и этот файл остается неизменным, пока не будет скомпилирована и поставлена пользователю новая версия, учитывающая изменения в операционных системах, аппаратуре и потребностях пользователей. При современных темпах развития индустрии программирования “статичные” приложения значительно замедляют процессы создания программных систем. Одно из решений этой проблемы состоит в том, чтобы разбить монолитное приложение на отдельные части, или компоненты. По мере развития технологии компоненты, составляющие приложение, могут заменяться новыми. Приложение более не является статичным, оно постепенно эволюционирует — исходные компоненты заменяются новыми. Из существующих компонентов легко создать и абсолютно новые приложения. Компонент подобен миниприложению: он поставляется пользователю как двоичный код, скомпилированный и готовый к использованию. Для того чтобы разбить монолитное приложение на компоненты, необходим мощный инструмент, которым и является COM.

COM (модель компонентных объектов) — определяет, как создавать динамически взаимосвязанные компоненты и клиенты, чтобы гарантировать возможность совместной работы, и устанавливает стандарт для конкретной архитектуры построения объектных приложений.

С возрастанием производительности и общего значения сетей увеличивается потребность в приложениях, состоящих из размещенных в сети компонентов. Компонентная архитектура помогает упростить процесс разработки подобных распределенных приложений. Использование данной архитектуры позволяет разработчику создавать приложения либо с учетом конфигурации сети (например, в ГИС центре ИНЦ выделен сервер и определены клиенты), либо без учета размещения компонентов. Для реализации компонентной архитектуры в рамках сети (как локальной, так и глобальной) предложена технология распределенной COM — DCOM (Distributed COM).

Существуют различные варианты коммутации компонентов с использованием интерфейса. Используя интерфейс COM/DCOM обеспечивается коммутация компонентов напрямую. Данный интерфейс включает в себя набор функций, которые реализуются компонентами и используются клиентами, т. е. представляет собой определенную структуру в памяти, содержащую массив указателей на функции. Другой подход, обеспечивающий большую гибкость связывания компонентов, называется “Автоматизация”. “Автоматизация” облегчает интерпретируемым языкам и макроязыкам доступ к компонентам COM, а также написание таких компонентов на этих языках. Особенностью “Автоматизации” является упор на проверку типов во время выполнения. Этот способ коммутации компонентов используют многие фирменные приложения Microsoft. Средства “Автоматизации” позволяют организовать иерархическую структуру объектов — методы и свойства могут возвращать указатели на интерфейсы других компонентов. “Автоматизацию” можно рассматривать как стандартный механизм, при помощи которого объект задает свои

свойства, методы и типы и предоставляет к ним доступ как через методы стандартного интерфейса IDispatch, так и методами обычных интерфейсов. “Автоматизация” является надстройкой над COM; компонент COM, реализующий интерфейс IDispatch, называется “Сервер Автоматизации”. Клиент COM, взаимодействующий с “сервером Автоматизации” через интерфейс IDispatch, называется “Контроллером Автоматизации”. Клиенты не вызывают функции сервера напрямую, вместо этого они используют функции-члены интерфейса IDispatch для неявного вызова функций “сервера Автоматизации”. Интерфейс IDispatch предоставляет клиентам и компонентам новый способ общения между собой. Вместо предоставления нескольких собственных специфичных интерфейсов для доступа к своим сервисам компонент может обеспечить доступ к этим сервисам через один стандартный интерфейс IDispatch.

Спецификация COM (DCOM) является основой таких технологий, как OLE и ActiveX, которые представляют собой всего лишь две разновидности реализаций указанных выше сервисов, построенных на их базе, эффективно “склеивающие” не зависящие от расположения объекты, переносимые между различными платформами, и не ограничивающиеся в выборе используемых языков программирования. OLE (object linking and embedding) — связывание и внедрение объектов. При связывании объекту указывается только ссылка на него, а при внедрении — компонент встраивается в объект, где выделяется место для его хранения, и т. д. Спецификация была распространена среди разработчиков в конце 1993 — начале 1994 гг. Позднее, в 1996 г., название изменилось на ActiveX и реализация была внедрена в Microsoft Internet Explorer.

Центральной технологией при разработке распределенной ГИС ИНЦ СО РАН является DCOM, реализованный на основе интерфейса IDispatch и метода CreateRemoteComObject для организации выполнения объекта на одном компьютере под управлением другого компьютера.

Базовым ПО настольных географических информационных систем в ГИС центре ИНЦ СО РАН являются MapInfo v4.1 и ARC/INFO v7.1.2. ГИС сервер представляет собой компьютер с базовой операционной системой MS Windows NT и набором программного обеспечения (для создания, отображения, редактирования карт), библиотек (для организации доступа клиентов в виде ActiveX компонентов), топоосновы региона в форматах 1:200 000, 1:1 000 000. Клиентское место — браузер стандартных HTML-страниц. При обработке браузером главной страницы сервера ГИС запускается программа, поддерживающая интерактивный интерфейс к программному обеспечению сервера.

Инструментальные ГИС предоставляют следующие возможности:

MapInfo (версия 4.5 и выше) обеспечивает как работу в локальной сети по технологии “клиент — сервер”, так и использование ее в качестве сервера OLE/COM/DCOM объектов (“Интегрированная Картография”) с интерфейсами к языкам MS Visual Basic, MS Visual C++, Delphi 3;

ARC/INFO (версия 7.1.2 и выше) — Open Development Environment (ODE) (“открытая среда разработчика”), которая позволяет разработчикам обращаться к функциональным возможностям ARC/INFO из других прикладных программ через специально созданные интерфейсы.

“Интегрированная Картография” — это механизм интеграции элементов MapInfo (окно Карты, Отчеты, Легенды и т. п.) в пользовательское приложение, обеспечивающий разработчиков простыми методами доступа к возможностям MapInfo из пользовательских приложений (связывание компонентов).

“Открытая Среда Разработчика” (ODE) представляет собой набор библиотек с од-

ной библиотекой для каждого стандартного модуля. ODE предоставляет доступ к функциональным возможностям ARC/INFO в виде набора подпрограмм и функций. ODE поддерживает те же команды и функции, которые доступны в традиционных модулях ARCEDIT, ARCPLOT, ARC GRID, а также через командную строку ARC/INFO. Версия ODE для Windows/NT включает OLE-оболочку для каждой динамически связанной библиотеки (DLL). В результате мы имеем три управляющих элемента ActiveX (ActiveX Controls) и два сервера объектов COM. ActiveX Controls включают в себя средства работы с картами ARCEDIT (Arcedit.ocx), ARCPLOT (Arcplot.ocx) и ARC GRID (Grid.ocx). Серверы объектов COM включают библиотеку функционального уровня ARC (Arc.dll) и уровня строкового массива (Strings.dll), которые возвращают информацию для среды разработки. ActiveX Controls помещаются в прикладную программу для получения доступа к функциональным возможностям ARCEDIT, ARCPLOT и ARC GRID.

Кратко опишем реализацию на примере MapInfo. Инициализация серверного объекта происходит следующим образом:

```
const IID_IMalloc: TGUID='{4A348528-2236-11D2-8990-004095D10D51}';
```

Определяется значение “глобально-уникального идентификатора” (GUID — global unique identifier), причем “глобальность” следует понимать буквально. Основной принцип заключается в том, что GUID однозначно определяет объект и интерфейс и представляет собой 16-байтовое число, которое всегда должно рассматриваться в целостности. GUID обычно записывается в виде последовательности из 32 шестнадцатеричных цифр, не обладающих самостоятельным значением в системном реестре Windows, для использования как через COM, так и другими программами клиента.

```
Var MapInfo : Variant;
```

```
MapInfo:=CreateRemoteComObject('62.76.23.145',IID_IMalloc) as IDispatch;
```

На указанной по IP-адресу машине происходит поиск и активизация объекта (MapInfo) в соответствии с указанным GUID.

Результатом выполнения является указатель MapInfo на интерфейс IDispatch, используемый далее для взаимодействия и вызова удаленных методов ГИС MapInfo, например:

```
1. MapInfo.RunMenuCommand(id_command);
```

где `id_command` — идентификатор команды. Позволяет вызвать стандартную команду ГИС MapInfo так, как если бы мы указали ее имя в списке меню (заметим, что клиент запрашивает выполнение команды в режиме on-line и удален от ее реализации).

```
2. MsgString:='Set Application Window'+sWinHand;
```

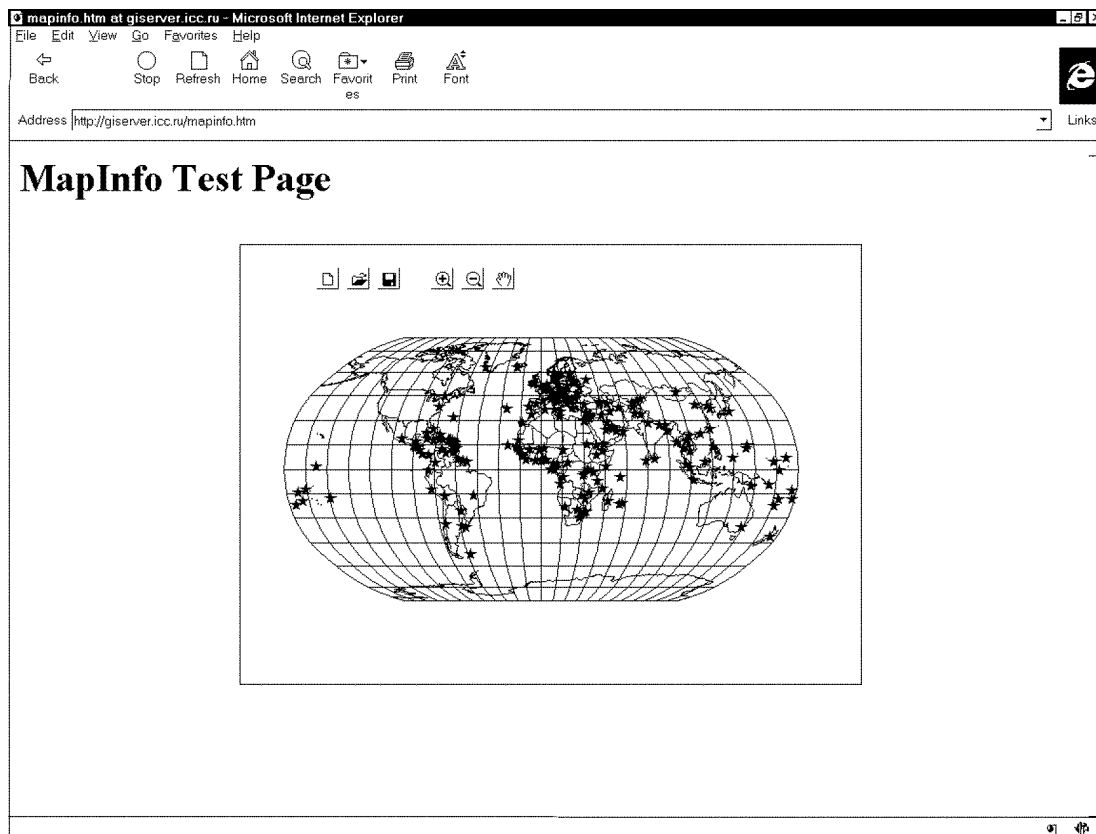
```
MapInfo.Do(MsgString);
```

Выполняет командную строку языка MapBasic так, как если бы мы вводили ее с клавиатуры. `MsgString:=MapInfo.Eval('SystemInfo(0)')`;

Вычисляет значение MapBasic-выражения и возвращает результат.

Заметим, что соответствующие функциональные возможности компонента MapInfo запрашиваются во время выполнения (технология позднего или динамического связывания). Позднее связывание является наиболее мощным и гибким, но, в отличие от раннего и ID-связывания, медленным технологическим методом динамического взаимодействия программных модулей для обмена функциональными возможностями.

Таким образом, показана возможность реализации удаленного доступа к картографическим ресурсам на основе описанных технологий. К достоинствам предложенного под-



хода следует отнести то, что используемый уровень интеграции применяемых компонентов обладает возможностью масштабирования, т. е. увеличение количества пользователей не влияет на функциональные возможности программного продукта; при использовании технологии OLE нет необходимости в написании специального программного обеспечения для взаимодействия между СУБД и ГИС. Основной проблемой при реализации данной системы являются обеспечение необходимого быстродействия, а также некоторая ограниченность, связанная с использованием в качестве серверной платформы операционной системы Windows NT, из-за чего возникает необходимость значительного расширения ресурсов ГИС сервера и использования в качестве клиентского места операционной системы Windows95/NT с браузером Internet Explorer.

Вышеизложенное позволяет говорить о перспективности используемых подходов. В настоящее время планируется дальнейшее расширение использования компонентов “черных ящиков”, позволяющих одной программе управлять выполнением другой посредством установки/чтения свойств объектов и вызова их методов; пока рынок указанных систем не перенасыщен подобными решениями и стандартами.

Поступила в редакцию 15 мая 1998 г.