

УДК 004.451.5:519.8

С.Ю. Золотов, Е.Л. Додолин

Методологические основы распределенной информационной системы мониторинга состояния окружающей среды

Рассматривается метод интеграции метеорологических данных состояния окружающей среды. Исходные данные накапливаются в своих собственных локальных базах данных, имеющих различную структуру их хранения. Техническая составляющая информационной системы обеспечивается технологией Java 2.

Ключевые слова: распределенная информационная система, Java 2 Enterprise Edition, мониторинг окружающей среды.

Проблема интеграции существующих систем хранения данных и обеспечение единого централизованного доступа к ним принимает все большую актуальность для Института мониторинга климатических и экологических систем (ИМКЭС) СО РАН (г. Томск). Институт производит сбор различных метеорологических данных, используя, как правило, автономные измерительные комплексы. Полученная таким образом информация сохраняется в индивидуальной базе данных. В общем виде данный процесс отображает рис. 1.

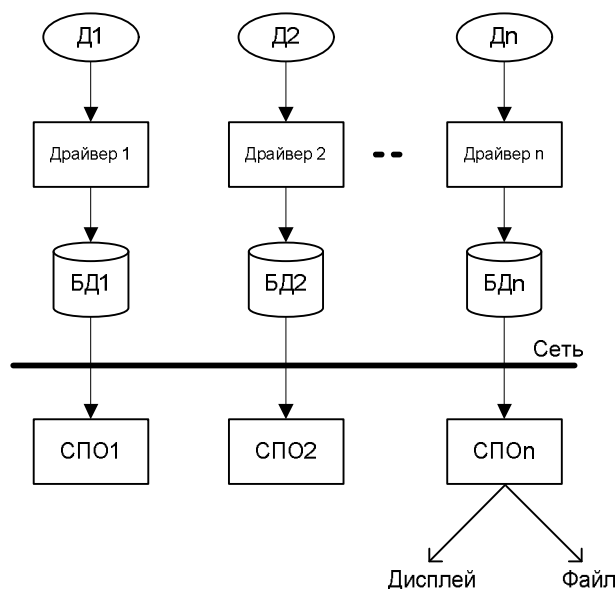


Рис. 1. Механизм получения метеорологических данных: СПО – специализированное программное обеспечение; БД – база данных; Д – первичный преобразователь информации (датчик)

Порядок и структура представления данных внутри базы являются индивидуальными для каждого измерительного комплекса. Ввиду отсутствия единой структуры возникают сложности при написании программного обеспечения для анализа и предоставления данных.

Чтобы объединить отдельные измерительные комплексы в единую распределенную систему, необходимо решить задачу интеграции различных структур представления данных [1, 2]. Теоретический подход интеграции структур D_i с целью представления данных в виде общей модели M сводится к созданию соответствующей функции преобразования F_i модели представления:

$$\begin{aligned}
 D_1 &\xrightarrow{F_1} M, \\
 D_2 &\xrightarrow{F_2} M, \\
 &\dots \\
 D_n &\xrightarrow{F_n} M.
 \end{aligned}$$

Дальнейшая работа с набором данных сводится к созданию методов доступа и анализа на основе общей модели M , что в свою очередь, позволяет использовать единый механизм обработки информации. Главная задача, требующая решения, заключается как раз в выборе средств и методов практической реализации преобразующей функции.

Основная часть программного обеспечения разрабатывалась на основе языка программирования высокого уровня Java 2. Для снижения материальных затрат на разработку и поддержку программного комплекса было принято решение использовать бесплатное программное обеспечение с открытым исходным кодом. На базе технологии Java 2 Enterprise Edition (J2EE) существует масса бесплатных продуктов, которые позволяют охватить сразу несколько ключевых аспектов:

- 1) снизить затраты на разработку и использование ПО;
- 2) отказаться от привязки к операционной системе;
- 3) использовать готовые технологии для решения стандартных задач управления ресурсами и обеспечения контроля доступа к этим ресурсам;
- 4) осуществлять взаимодействие с другими системами, используя web-сервисы (SOAP), ПОР-, CORBA-, и RMI-протоколы.

Возможности технологии J2EE позволяют работать с СУБД наиболее известных производителей, таких как MySQL, Oracle, PostgreSQL и других без изменения логики компонентов самого приложения.

К разрабатываемой на практике системе предъявляются следующие основные требования: низкая себестоимость, высокая надежность в эксплуатации, безопасность хранения и доступа к данным, простота в обслуживании и эксплуатации, обеспечение максимально простого механизма расширения и модернизации системы.

Платформа J2EE представляет собой среду для создания и установки клиент-серверных приложений, ориентированных как на внутренние сети, так и на использование в сети Интернет. Наиболее широкое распространение данная технология получила в коммерческих сферах, таких как управление корпоративными ресурсами, регулирование финансовых потоков, Интернет-коммерция и т.п. Функциональной единицей данной технологии, выполняющей необходимую рабочую логику, является Enterprise Java Bean (EJB) компонент. Java Naming and Directory Interface (JNDI) обеспечивает доступ к EJB, используя соответствующие им уникальные имена. По сути компонент EJB является связующим звеном между областью хранения и представления данных. Обеспечивает среду выполнения J2EE-компонентов сервер приложений. Детально познакомиться с технологией и ее концепцией можно в [3–6].

Из многочисленных достоинств технологии J2EE следует отметить те, которые способствуют решению задачи создания механизма интеграции БД, – это отличная масштабируемость приложений, встроенный механизм поддержки транзакций, надежная система обеспечения безопасности доступа к данным и простота реализации конечного продукта.

В общем виде функциональная схема части информационной системы, отвечающей за преобразование индивидуальной модели данных в обобщенную, приведена на рис. 2. Для реализации преобразующей функции F_i используется драйвер, представляющий собой EJB компонент. Каждый такой драйвер реализует один и тот же интерфейс Java, тем самым достигается однородность выходных данных. Подобный механизм широко используется в решении задач стандартизации представления данных и операций над ними. На основе такого шаблона создаются, например, JDBC-ODBC драйвера к базам данных различных производителей. По мере необходимости драйвер (компоненты EJB) можно добавлять или удалять, не внося изменений в общую структуру системы. Идентификация драйвера происходит по JNDI имени компонента EJB.

Программная реализация логики драйвера заключается в создании определенных SQL запросов к соответствующей БД на основе входных данных и формировании по результатам этих запросов ответа в форме, определенной Java интерфейсом. Форма запросов определяется действительной моделью хранения информации в базе.

Управляет драйверами и их конфигурирует контроллер БД, реализованный как отдельное EJB приложение 1. Его функция заключается в регистрации новых драйверов, удалении имеющихся, изменении конфигурационной информации и предоставлении драйверов клиентским приложениям. Для хранения конфигурации драйверов контроллер использует локальную БД (ЛБД). Клиент, желающий обратиться к своей БД, обязан предоставить контроллеру уникальное имя источника данных, зарегистрированное в ЛБД контроллера, по которому определяется JNDI имя EJB-компонента драйвера.

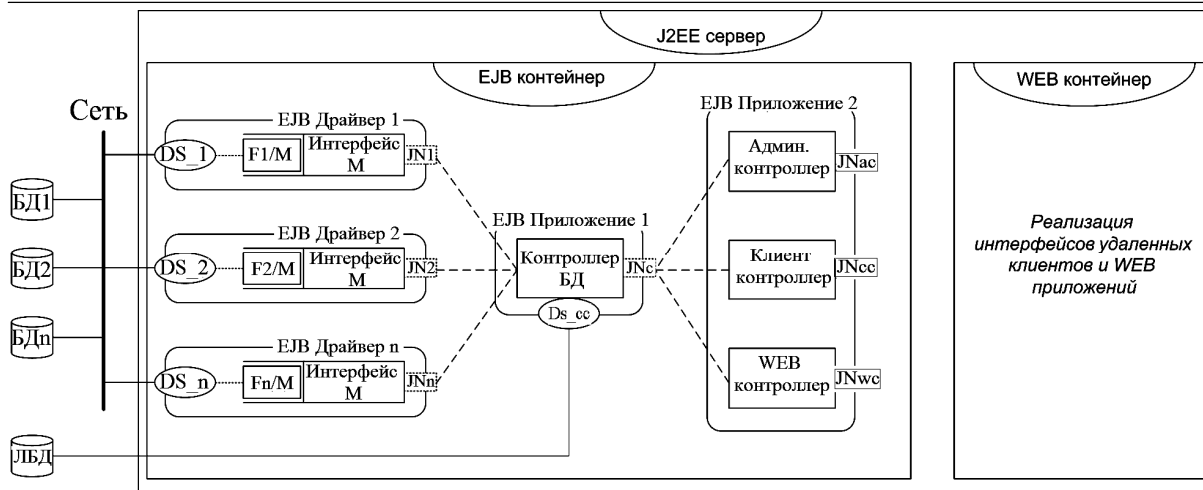


Рис. 2. Функциональная схема преобразующей части ИС: БД – база данных, ЛБД – локальная БД; Ds_x – дескриптор связи, отвечающий за соединение с БД; Fn/M – приложение, реализующее конвертацию данных

Приложение 2 отражает следующий за контроллером БД этап обработки данных. Оно может включать в себя любое необходимое число EJB-компонентов, которые реализуют обработку данных, необходимую для конкретного клиента. В данном примере на рис. 2, приложение 2 состоит из трех компонентов. Контроллер администрирования реализует функциональность управления драйверами и предназначен для работы исключительно под управлением администратора системы. Контроллер удаленного клиента реализует методы управления потоками данных (чтение/запись) по Интернету применительно к клиентским прикладным программам, а также WEB-контроллер, предоставляющий механизм безопасного доступа к данным для создания Интернет-приложений.

Контроллеры приложения 2 используют уже стандартное представление данных, поэтому их не требуется переписывать в случае появления новых баз и драйверов или при удалении имеющихся. Единственное, что требуется, это составление корректных запросов к соответствующему драйверу после обработки форм пользовательских интерфейсов.

Отметим важную особенность платформы J2EE, касающуюся взаимодействия с БД. Механизм доступа к базам построен по принципу декларирования соединения в специальном дескрипторе, представляющем собой XML-файл, и обращении к БД через присвоенное данному соединению специальное имя. Дескриптор выполняет функции инициализации необходимых драйверов, аутентификации на сервере БД и определения необходимой конфигурации соединения. Создавая приложение, работающее с БД, отпадает необходимость решать проблемы, связанные с соединением и спецификой поставляемой БД, эту функцию берет на себя сервер приложений. Также сервер отвечает за выполнение транзакций, предоставляя пользователю выбирать нужные ему режимы.

Предложенная методика интеграции данных имеет ряд следующих преимуществ.

Низкая себестоимость создания системы обусловлена применением технологий открытого типа для реализации модели. Использование технологий на основе Java2 позволяет полностью перейти на бесплатное программное обеспечение, используя операционную систему на базе Linux и СУБД, такие как MySQL или PostgreSQL.

J2EE в настоящее время приобрела большую популярность в Интернет-коммерции, что позволило разработчикам данной технологии сконцентрировать свое внимание на увеличении надежности среды управления данными и обеспечении безопасности доступа к ресурсам. Встроенные возможности (шифрование пароля, контроль доступа, аудит) и дополнительные модули обеспечивают полную безопасность приложений на протяжении всего процесса разработки и внедрения.

Используя средства управления проектами Java 2 (например, на основе технологий Maven 2 и SVN), удалось полностью автоматизировать процесс сборки и компоновки J2EE-приложений. Упростить процедуру написания дескриптора позволяет регламентированный набор правил описания метаданных, XML-тэгов и связей.

Однако реализация описанным методом имеет и недостаток. Очевидно, что, создавая промежуточный слой прохождения информации, мы заведомо теряем в ее скорости обработки. Величина снижения производительности, таким образом, напрямую зависит от технологии и выбранных алгоритмов приложений. Тонким местом в предложенной

выше схеме являются каналы взаимодействия между компонентами и БД. Частично решить эту проблему можно при соответствующей настройке сервера приложений для оптимизации потенциальных потерь в скорости обработки информации.

В рамках концепции J2EE оказалось возможным построить систему, удовлетворяющую приводимым выше требованиям. Открытость и популярность технологии позволяют минимизировать затраты на ее внедрение в рабочий процесс практически любой отрасли. Ориентация преимущественно на сетевое использование подразумевает богатый арсенал средств для реализации качественных Интернет-приложений.

Литература

1. Чертовской В.Д. Базы и банки данных: учеб. пособие. – М.: Изд-во МГУП, 2001. – 219 с.
2. Базы данных: курс лекций / Под ред. Е.Г. Слекеничс. – Обнинск: Принтер, 2002. – 139 с.
3. Sriganesh R.P. Mastering Enterprise JavaBeans 3.0 / R.P. Sriganesh, G. Brose, M. Silverman. – USA: Wiley Publishing, Inc., 2006. – 685 p.
4. Кассем Н. Разработка бизнес-приложений с помощью Java 2. – М.: Лори, 2002. – 284 с.
5. Дейтел Х.М. Технологии программирования на Java 2 / Х.М. Дейтел, П.Дж. Дейтел, С.И. Сантри. – М.: ООО «Бином-Пресс», 2003. – 560 с.
6. Ноутон П., Java 2: наиболее полное руководство / П. Ноутон, Г. Шилдт. – СПб.: ВHV, 2007. – 1072 с.

Золотов Сергей Юрьевич

Канд. техн. наук, доцент каф. АСУ ТУСУРа,
м.н.с. Института мониторинга климатических и экологических систем (ИМКЭС) СО РАН, г. Томск
Тел.: 8-3822-70-15-36
Эл. почта: zolotov@asu.tusur.ru

Додолин Евгений Леонидович

Аспирант ИМКЭС СО РАН
Эл. почта: johni5@imces.ru

Zolotov S.Yu., Dodolin E.L.

Methodological background to distributed information system of environmental monitoring

In the article the method of integration of any meteorological data of the environment state is considered. Input data are stored in the own local databases having various storage structures. The technical component of the information system is provided by the Java 2 technology.

Keywords: Distributed information system, Java 2 Enterprise Edition, environmental monitoring.
