

УДК 621.311.001.57

РАБОТА СО СЛОЖНЫМИ СТРУКТУРАМИ ДАННЫХ РСУБД В СРЕДЕ LABVIEW

Попов Д.А.¹

¹ФГБОУ ВПО «Пермский Национальный исследовательский политехнический университет», Пермь, Россия (614000, Пермь, ул. Комсомольский проспект, 29), e-mail: rector@pstu.ru

В статье рассмотрены стандартные способы взаимодействия среды LabView с РСУБД и недостатки их использования при работе со сложными структурами данных. Предложен метод взаимодействия системы LabView с РСУБД на основе техники объектно-реляционного маппинга. Показан порядок реализации и преимущества использования предложенного метода в среде LabView. Произведена оценка скорости получения данных с использованием ORM Entity Framework в среде LabView с помощью вызова сгенерированной dll через компонент constructor node. Даны рекомендации по использованию метода получения данных на базе ORM в среде LabView. Предложен способ повышения скорости получения данных с использованием dll с функционалом ORM. Показаны особенности и перспективы использования объектно-реляционного маппинга в среде LabView.

Ключевые слова: временные ряды, ORM, LabView, EntityFramework, dll, РСУБД.

DEALING WITH COMPLEX DATA STRUCTURES OF RELATIONAL DATABASE IN LABVIEW

Popov D.A.¹

¹Perm National Research Polytechnic University, Perm, Russia (614000, Perm, street Komsomolskij prospekt, 29), e-mail: rector@pstu.ru

The article describes the standard ways of interaction of LabView and relation database and disadvantages of using them when dealing with complex data structures. The method of interaction with relational database based on object-relational mapping technology, shows how the implementation and benefits of the proposed techniques. The rate estimation of production data using ORM Entity Framework in LabView has been realized by calling the generated dll through the component constructor node. The recommendations for using the acquisition method based on ORM LabView environment are given. The method for increasing the speed of data acquisition using dll with functional ORM is proposed. The features and the prospects of using object-relational mapping in the LabView environment is represented.

Keywords: time series, ORM, LabView, EntityFramework, dll, relational database.

Введение

При решении научных и технических задач, связанных с измерениями или компьютерным моделированием, необходима работа со сложной структурой данных. Например, такая необходимость возникает при испытании газотурбинных установок большой мощности [1]. Для реализации подобных задач часто используется среда LabView, в которой разработчик имеет возможность использовать инструменты взаимодействия с базами данных на основе интерфейсов ODBC, ADODB и OLE DB с вариантами подключения ODBC, Promt, UDL (рис. 1) [4].

Таким образом, для получения данных из БД необходимо проходить следующие стадии работы с данными:

- 1) открыть подключение к базе данных;

- 2) сформировать запрос;
- 3) осуществить вызов API или COM (выполнить запрос);
- 4) произвести обработку (парсинг) полученного результата;
- 5) закрыть подключение.

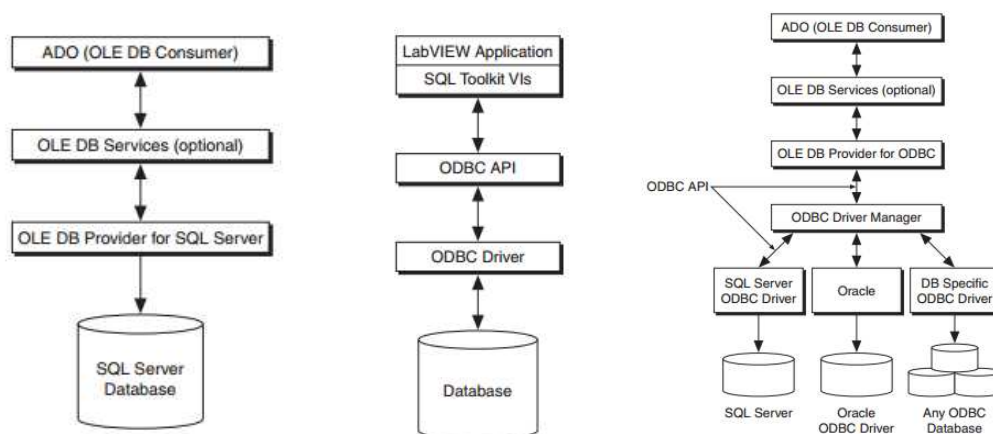


Рис. 1. Взаимодействие LabView с базой данных SQL Server на основе соответственно OLE DB, ODBC и ADO

В сложных распределенных системах для обеспечения модульности и повторного использования необходимо выносить логику работы с базами данных в отдельные подпроекты / виртуальные приборы. В этом случае должны присутствовать виртуальные приборы, обеспечивающие базовые однотабличные функции создания, редактирования, обновления и удаления данных (CRUD-функции) [7], а также специализированные CRUD-функции, например, изменение значения одного объекта на основании значений другого или ручное каскадное удаление данных. При этом сложные команды могут быть сохранены как хранимые процедуры. В таком случае общее число виртуальных приборов останется значительным, а следовательно, сложность сопровождения программного продукта повысится, как и трудозатраты на разработку данной программной прослойки. Так, при работе с РСУБД, содержащей 12 сущностей, в среднем необходимо разработать 48 простых CRUD-функций, а также дополнительные функции, общее число которых зависит от сложности проекта и может достигать нескольких сотен, что достаточно трудоемко и тяжело в сопровождении.

При реализации проекта с использованием встроенных средств взаимодействия с РСУБД затруднительно проведение профилирования и тестирования разрабатываемого приложения, поскольку функции взаимодействия с БД жестко прописаны в коде. Для проведения

независимого профилирования необходимо создать заглушки для всех функций работы с БД – пустые или элементарные функции, заменяющие функции взаимодействия с данными.

Кроме того, на протяжении всей стадии разработки приложения необходимо сопровождение продукта специалистом по базам данных.

Для упрощения процесса разработки приложения необходимо обеспечить возможность работы с базой данных через более высокоуровневый интерфейс – прослойку между БД и приложением. Причем данная прослойка должна выполнять все служебные функции: открытие, закрытие соединения с БД, конфигурирование кэширования – определение способов получения данных, формирование запросов, исполнение запросов, то есть обеспечить инкапсуляцию работы с базами данных.

Цель данной статьи – разработать типизированный и интуитивно понятный способ взаимодействия программы LabView с реляционными базами данных, позволяющий сократить затраты на разработку приложений со сложной структурой данных и обеспечивающий возможность тестирования и профилирования приложения, с предоставлением инструментов кэширования данных и обеспечением высокой степени инкапсуляции работы с базами данных.

Решение задачи

При работе со сложными структурами данных в объектно-ориентированных языках программирования целесообразно использовать технологию программирования, которая связывает базы данных с концепциями объектно-ориентированных языков программирования, создавая «виртуальную объектную базу данных» – объектно-реляционный маппинг (ORM). Применение объектно-реляционного маппинга позволяет значительно ускорить разработку приложения, а также обеспечить инкапсуляцию, кэширование и фильтрацию данных [2].

Среда LabView не содержит встроенных инструментов, предоставляющих функции объектно-реляционного маппинга. Для получения возможностей ORM в среде LabView необходимо использовать сторонние разработки ORM, реализованные на базе платформы, отличной от LabView. Так как среда разработки LabView содержит инструменты интеграции со средой Visual Studio, то выберем платформу Microsoft .NET, базу данных Microsoft SQL Server и рекомендованную компанией Microsoft технологию ORM Entity Framework [8].

Реализация технологии Entity Framework представляет собой набор классов, которыми оперирует среда .net framework для обеспечения получения данных из БД на основе контекстов и файлов конфигурации [5].

Взаимодействие среды LabView и платформы .NET может быть осуществлено на базе вызовов dll в среде LabView.

Для создания dll на базе платформы .NET необходимо создать новую сборку, обеспечивающую подключение к базе данных на основе технологии Entity Framework. При реализации Entity Framework в данной сборке должны находиться сервисные (внутренние) классы Entity Framework, классы, соответствующие структуре базы данных (модель сущностей), ссылки на соответствующие файлы dll и файл конфигураций, содержащий информацию о работе Entity Framework, такую как строка соединения – строка, описывающая настройки подключения к базе данных и другую.

При переносе строки соединения из конфигурационного файла в код программы появляется возможность скомпилировать и использовать сборку, подключив ее к проекту LabView, используя инструмент constructor node среды LabView [3].

Таким образом, возможно работать непосредственно с моделью сущностей Entity Framework, получая и вызывая соответствующие свойства и методы. Например, имея связь сущностей устройство – параметры устройства, можно вызвать свойство модели – список параметров данного устройства для конкретного устройства.

При наличии часто используемых запросов к базе данных данные запросы можно описать новой функцией и поместить их в класс dll, причем такого рода функции могут использовать возможности платформы .net framework 3.5, например, лямбда-выражения. К примеру, можно дополнить класс Device методами получения информации об устройстве по коду устройства `GetDeviceByID(int DeviceID)`, получения информации об устройстве по определенному условию `GetDeviceByCondition(string Condition)`. Некоторые методы, критичные к скорости исполнения и обладающие высокой сложностью запросов, могут взаимодействовать с базой данных напрямую, формируя запросы и исполняя их на основании входных данных, без использования технологии ORM.

Для согласованной работы среды LabView и предлагаемой dll необходимо четкое соответствие модели сущностей dll и модели РСУБД. Для этого предлагается использовать подход Code-first для первичной разработки классов (модели базы данных) и создания базы данных на основании миграций и дальнейшего переноса структуры классов в приложение LabView. Использование технологии Code-first позволяет создать модель РСУБД, точно совпадающую с описанием объектов, а также автоматически создать экземпляр базы данных [6]. Следовательно, приложения будут работать и взаимодействовать друг с другом и с РСУБД с полностью идентичными структурами – объектными моделями.

Для достижения поставленных целей необходимо разработать систему LabView на базе техники ORM. Данная работа может состоять из нескольких этапов:

- 1) разработка объектной модели;
- 2) создание сборки и подключение Entity Framework;
- 3) создание классов в рамках контекста в сборке на основании объектной модели;
- 4) миграция контекста и создание базы данных;
- 5) построение сборки;
- 6) использование сборки в среде LabView.

После выполнения предложенных этапов появится возможность использовать предлагаемую технику ORM и инкапсулировать процесс работы с данными, что в свою очередь даст возможность управления правами доступа на уровне приложения, кэширования, фильтрации данных и значительного сокращения количества кода и временных затрат на разработку и сопровождение приложения. Использование ORM позволяет использовать наследование – инструмент объектно-ориентированного приложения, причем появляется возможность использовать любую схему наследования для одного и того же кода.

Пример использования ORM в среде LabView

Для примера реализуем добавление записи о новом устройстве в базу данных с помощью встроенных средств работы с БД LabView (рис. 2) и с помощью предлагаемой техники (рис. 3).

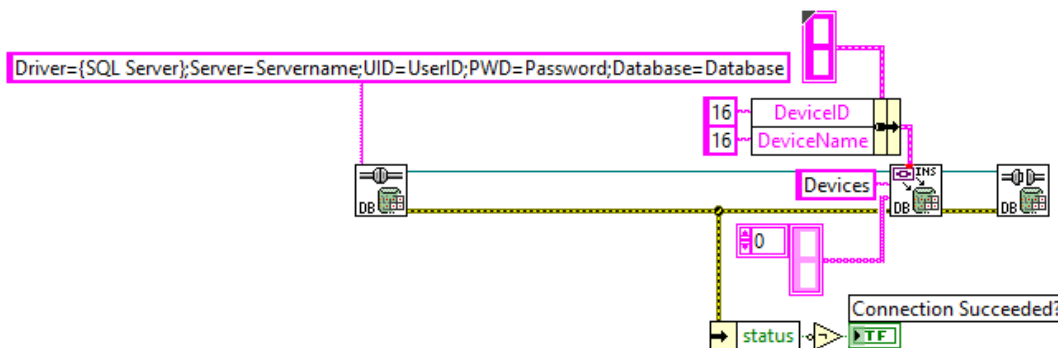


Рис. 2. Добавление записи о новом устройстве в базу данных с использованием упрощенной формы работы с базой данных, скрывающей формирование и выполнение запроса

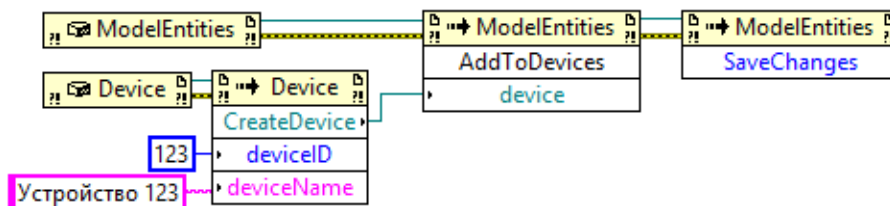


Рис. 3. Добавление записи о новом устройстве в базу данных с помощью техники ORM

Необходимо отметить строгую типизацию и интуитивно понятный интерфейс работы с БД по технике объектно-реляционного маппинга, также следует отметить инкапсуляцию работы с базой данных – инженер работает с базой данных как с объектами, при этом ему не нужно заботиться об открытии, закрытии соединения, о низкоуровневых деталях работы с базой данных.

Разработанная dll может быть повторно использована в других приложениях, требующих работы с аналогичными данными, а также соответствующим образом дополнена для разграничения прав доступа к БД по ролям со стороны приложения.

Характеристики работы программы.

Для обеспечения высокой скорости работы программы с использованием предлагаемой техники необходимо осуществить загрузку используемых dll. Для этого достаточно проинициализировать dll, подгрузить их в оперативную память. В среднем это занимает порядка 1 с.

Считывание конфигурации устройства, зависящей от двух связанных сущностей при средней заполненности таблиц 1000 значениями в базе данных Microsoft SQL Server 2008 R2 на платформе asp.net 3.5 с использованием предлагаемой библиотеки dll, реализованной на базе ORM-технологии EntityFramework v5, составляет порядка 30 мс. В то же время взаимодействие с базой данных стандартными средствами из палитры connectivity по интерфейсу ADOdb составляет порядка 28 мс (без учета времени открытия соединения). С учетом времени открытия соединения общее время доступа к данным увеличивается и составляет 50 мс.

После сравнительного временного анализа получения данных по предлагаемой технике объектно-реляционного маппанга на основании Entity Framework с использованием dll и по интерфейсу ADOdb, можно сделать следующие выводы.

1. Использование предлагаемой техники показывает в простых запросах скорость равную и даже превышающую скорость выполнения прямых запросов к базе данных
2. При запросах, требующих несколько операций соединения join, предлагаемая техника генерирует большее число обращений к базе данных, что снижает время работы с данными, но при этом все полученные результаты абсолютно предсказуемы [9].
3. Подтверждено, что использование dll с функционалом ORM Entity Framework позволяет увеличить скорость работы с базой данных посредством кэширования, которое настраивается в подключаемой сборке благодаря использованию атрибутов Entity.
4. Существует возможность использования среды Visual Studio как средства отладки используемой dll посредством присоединения к выполняемому процессу.

Перспективы развития технологии и способы применения.

Учитывая, что предлагаемая техника на базе ORM не заменяет стандартный способ работы с базами данных, а лишь дополняет его, в случае, когда требуется критичная по времени работа с данными, можно использовать оба подхода одновременно.

Поэтому следующим возможным шагом в развитии данной техники является объединение используемых в приложении LabView классов для реализации системы и классов PCУБД. Это возможно при объектно-ориентированном подходе к разработке приложения LabView, а также при наличии дополнительных адаптеров, настраиваемых над каждым классом и позволяющих напрямую работать с объектами ORM через интерфейс классов LabView.

В этом случае ORM придает базе данных объектность и инкапсулированность, а концепция языка программирования LabView как раз заключается в инкапсулированности и коробочной реализации требуемых компонентов.

Таким образом, используя подход интеграции Entity Framework в dll LabView возможно разработать механизм синхронизации системы управления версиями проекта LabView с данными о миграциях, что расширит возможности LabView и позволит вызывать функции получения данных прямо из классов системы LabView.

Список литературы

1. Попов Д.А., Шмидт И.А. Разработка функциональной структуры программного комплекса испытаний газотурбинных установок мощностью до 40 МВт // Научные исследования и инновации. – 2012. – Т. 6. – С. 264–270.
2. Ambler S.W. Mapping objects to relational databases // Softw. Dev. 1995. Vol. 3. P. 63–72.
3. Bitter R., Mohiuddin T., Nawrocki M. LabVIEW: Advanced Programming Techniques. Boca Raton: CRC Press/Taylor & Francis Group, 2007.
4. Instruments N. Database Connectivity Toolkit User Manual [Электронный ресурс]. – Режим доступа: <http://www.ni.com/pdf/manuals/321525c.pdf> (дата обращения: 19.12.2013).
5. Lerman J. Programming Entity Framework: DbContext. O'Reilly Media, 2012.
6. Lerman J. Programming Entity Framework. O'Reilly, 2010.
7. Martin J. Managing the Data-base Environment. New Jersey: Prentice-Hall, 1983.
8. Microsoft. Entity Framework [Электронный ресурс]. – Режим доступа: <http://msdn.microsoft.com/en-us/data/ef.aspx> (дата обращения: 19.12.2013).

9. Performance measuring test suite [Электронный ресурс]. – Режим доступа: <http://ormbattle.net>
(дата обращения: 19.12.2013).

Рецензенты:

Аликин В.Н., д.т.н., профессор, Научно-исследовательский институт управляющих машин и систем, г. Пермь.

Хрипченко С.Ю., д.т.н., профессор, Пермский государственный национальный исследовательский университет, г. Пермь.