

6. Review of the information on robotic complexes. Available at: http://mchs.gov.by/_modules/_cfiles/files/obzor_o_robotizirovannih_kompleksah.pdf (accessed 11 November 2014). (In Russ.)
7. Review of information on FLANN. Available at: <http://www.cs.ubc.ca/research/flann/> (accessed 30 October 2014). (In Russ.)
8. Chernukhin Yu. V., Dolenko Yu. S., Butov P. A. Neyrosetevoy podkhod k reshetsiyu zadachi lokalno navigatsii intellektualnymi mobilnymi robotami v usloviyakh, priblizhennykh k realnoy srede [A neural network approach to the problem locally reshetsiyu mind-trivial navigation of mobile robots in conditions close to the real environment]. *Izvestiya Yuzhnogo federalnogo universiteta. Tekhnicheskie nauki* [Southern Federal University. Technical Sciences], 2013, no. 5 (142), pp. 80–84.
9. Bay H., Ess A., Tuytelaars T., Van Gool L. Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*, 2008, June, vol. 110, iss. 3, pp. 346–359.
10. Bonin-Font F., Ortiz A., Oliver G. Visual Navigation for Mobile Robots: a Survey. *Journal of Intelligent and Robotic System*, 2008, pp. 263–296.
11. Bota S. Camera motion estimation using monocular and stereo-vision. *Intelligent Computer Communication and Processing*, 2008, pp. 275–278.
12. Bradski G., Kaehler A. *Learning OpenCV*. O'Reilly Media, 2008. 576 p.
13. Forsyth D. A., Ponce J. Computer Vision: A Modern Approach. *Robotics and Autonomous Systems*, 2006, vol. 11, no. 3, pp. 84–92.
14. Juan L., Gwun O. A Comparison of SIFT, PCA-SIFT and SURF. *International Journal of Image Processing (IJIP)*, 2009, vol. 3, no. 4, pp. 143–152.
15. Kefer M., Kubinger W. Evaluation of Kinect Depth Sensor for Use in Mobile Robotics. *Annals of DAAAM for 2011 & Proceedings of the 22nd International DAAAM Symposium*. Vienne, 2011, vol. 22, no. 1.
16. k-d tree. Available at: http://en.wikipedia.org/wiki/K-d_tree (accessed 30 October 2014).
17. Marius Muja, David G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. *VISAPP International Conference on Computer Vision Theory and Applications*, 2009, pp. 331–340.
18. Ramisa A., Vasudevan S., Aldavert D., Toledo R., Lopez de Mantaras R. Evaluation of the SIFT Object Recognition Method in Mobile Robots. Available at: http://www.iia.csic.es/~mantaras/sift_eval.pdf. (accessed 30 October 2014)

УДК 004.4

ПРОЕКТИРОВАНИЕ РАСПРЕДЕЛЕННЫХ БАЗ ДАННЫХ В ГИБРИДНОЙ ОБЛАЧНОЙ ИНФРАСТРУКТУРЕ НА ОСНОВЕ ОБЪЕКТНО-РЕЛЯЦИОННОГО ОТОБРАЖЕНИЯ

Статья поступила редакцию 17.11.2014, в окончательном варианте 06.12.2014.

Лукьянчиков Олег Игоревич, преподаватель, Московский государственный университет приборостроения и информатики, 107996, Российская Федерация, г. Москва, ул. Стромынка, 20, e-mail: luk-it6@yandex.ru

Никольчев Евгений Витальевич, доктор технических наук, профессор, проректор по научной работе, Московский технологический институт, 119334, Российская Федерация, г. Москва, Ленинский пр-т, 38а; профессор Высшей школы системного инжиниринга, Московский физико-технический институт (государственный университет), 141700, Российская Федерация, Московская обл., г. Долгопрудный, Университетский пер., 9, e-mail: e_nikulchev@mti.edu.ru

Паяин Семен Владимирович, кандидат технических наук, Московский технологический институт, 119334, Российская Федерация, г. Москва, Ленинский пр-т, 38а, e-mail: sadsema@gmail.com

Плужник Евгений Владимирович, первый проректор, Московский технологический институт, 119334, Российская Федерация, г. Москва, Ленинский пр-т, 38а; аспирант, Московский государственный университет печати имени Ивана Федорова, 127550, Российская Федерация, г. Москва, ул. Прянишникова, 2А, e-mail: e_pluzhnik@mti.edu.ru

Одной из важных задач в области использования облачных сервисов сейчас является задача проектирования специализированных систем управления данными. Это особенно актуально для гибридных систем, в которых данные расположены и в общем, и частном облаке. Обеспечение функций мониторинга выполнения запросов, диспетчеризации и обработки должно быть надежно программно реализовано и являться неотделимой частью системы. Для обеспечения этих функций предложено использовать объектно-реляционные отображения (Object-Relational Mapping, ORM). Статья посвящена изложению подхода к проектированию баз данных для информационных систем, размещенных в гибридной облачной инфраструктуре. Приведен пример разработки ORM библиотеки. Использование систем виртуализации и разных каналов связи, не дает возможности провести оценку параметров эффективности алгоритмов и скорости выполнения запросов, поскольку это требует и экспериментального исследования и возможности оперативной модификации методов обработки. Важная функция, реализуемая с помощью ORM, дает возможность изменения любого из методов объекта без изменения архитектуры системы и полного разбора (анализа) программного кода. Предложенный подход также позволяет реализовать идеи использования обратных связей для программного конфигурирования доступа к данным и обработки запросов, что обеспечит безопасность доступа к данным и контроль их целостности.

Ключевые слова: облачные технологии, базы данных, объектно-реляционные отображения, проектирование баз данных, управление данными, облачные сервисы, облачная инфраструктура, объектно-ориентированный подход, программно-конфигурируемые системы, гибридное облако

USING OBJECT-RELATIONAL MAPPING TO CREATE A DISTRIBUTED DATABASE IN A HYBRID CLOUD INFRASTRUCTURE

Lukyanchikov Oleg I., lecturer, Moscow State University of Instrument Engineering and Computer Science, 20 Stromynka St., Moscow, 107996, Russian Federation, e-mail: luk-it6@yandex.ru

Nikulchev Evgeniy V., D.Sc. (Engineering), Professor, Vice-rector for scientific work, Moscow Technological Institute, 38a Leninskiy Avenue, Moscow, 119334, Russian Federation; Professor of High School of System Engineering, Moscow Institute of Physic and Technology, 9 Institutskiy per., Dolgoprudny, Moscow region, 141700, Russian Federation, e-mail: e_nikulchev@mti.edu.ru

Payain Semen V., Ph.D. (Engineering), Moscow Technological Institute, 38a Leninskiy Avenue, Moscow, 119334, Russian Federation, e-mail: sadsema@gmail.com

Pluzhnik Evgeniy V., First vice-rector, Moscow Technological Institute, 38a Leninskiy Avenue, Moscow, 119334, Russian Federation; post-graduate student, Moscow State University of Printing Arts, 2A Prynishnikov St., Moscow, 127550, Russian Federation, e-mail: e_pluzhnik@mti.edu.ru

One of the challenges currently problems in the use of cloud services is the task of designing of specialized data management systems. This is especially important for hybrid systems in which the data are located in public and private clouds. Implementation monitoring functions querying, scheduling and processing software must be properly implemented and is an integral part of the system. To provide these functions is proposed to use an object-relational mapping (ORM). The article devoted to presenting the approach of designing databases for information systems hosted in a hybrid cloud infrastructure. It also provides an example of the development of ORM library. The use virtualization and various communication channels, makes it impossible to assess the effectiveness of the algorithms and parameters query performance. Therefore, it requires experimental investigation and the possibility of rapid modification processing methods. Important function realized by using ORM, makes it possible to change any of the object's method without changing the system architecture and full parsing code. The proposed approach also allows for the idea of using feedback to the software configuration of data access and query processing, to ensure the security of data access and integrity control.

Keyword: cloud computing, database, object-relational mapping, database design, data management, cloud services, cloud infrastructure, object-oriented, software-configurable system, hybrid cloud

В процессе информатизации общества возрастает роль методов обеспечения эффективности хранения, доступа и обработки данных, в том числе и с использованием облачных хранилищ. Одним из новых направлений исследований и разработок является применение «гибридных» облачных хранилищ, использующих как удаленное хранение данных, так и «локальные» вычислительные ресурсы. Однако ряд вопросов по этому направлению остаются исследованными недостаточно полно. Поэтому целью данной работы был комплексный анализ общих принципов проектирования инфраструктуры эффективных гибридных облачных систем.

Введение в проблематику исследований. Современные приложения оперируют с большими объемами данных, которые могут находиться в различных хранилищах. Стремительно развиваются облачные технологии и облачные хранилища данных, которые имеют преимущества по производительности за счет распараллеливания вычислений; использования технологии виртуализации; масштабирования вычислительных ресурсов; организации доступа к данным через веб-интерфейс. Поэтому актуальной является задача переноса существующих систем и баз данных (БД) в облако.

Сейчас многие разработчики и пользователи озабочены вопросами полноценного использования возможностей облачных сервисов. Однако миграция существующих систем в облако пока только создает проблемы. Вопросы обеспечения безопасности доступа к данным и гарантирования уровня обслуживания могут быть решены путем использования гибридного облака – часть данных (обработка запросов, к которым требует больших вычислительных затрат, а сами данные не являются конфиденциальными) размещается в общем (публичном) облачном сервисе, а остальные данные – в частном (приватном) облаке или в локальной сетевой инфраструктуре [6]. Однако для этого случая не разработаны специализированные принципы проектирования облачных систем. Данная задача в теоретическом плане была рассмотрена в работах [4, 8, 14]. Известны ее решения для отдельных приложений [2, 3, 10, 12, 19].

Сложность построения методик проектирования распределенных баз данных в гибридном облаке, что невозможно для облака оценить параметры алгоритмов и выполнения запросов: в каждом случае покупаемое (приобретаемое) количество облачных ресурсов, таких как виртуальные машины, различно; неизвестны маршруты и характеристики каналов связи. Вопросы оптимизации объемов и видов ресурсов для являются отдельной, важной задачей. Поэтому в настоящее время, в отсутствии разработанных общих принципов и методик, эксперименты представляются единственным способом исследования эффективности проектных решений для рассматриваемой области исследований.

Авторами был создан экспериментальный стенд (ЭС), имитирующий работу с гибридными хранилищами. Сам стенд и некоторые экспериментальные результаты, полученные с его помощью, описаны в [5, 15, 16]. Используемое программное обеспечение VMware vCloud позволяет организовывать облачные вычисления на всех уровнях. Для создания облака в ЭС на двух серверах используется VMware ESXi. Установлены система управления VCenter, приложение VMware vCloud Director. В ЭС задействовано более 15 физических коммутаторов Cisco 29 серии и маршрутизаторов 26 и 28 серии, а также виртуальные коммутаторы Nexus. Система на базе ЭС позволяет имитировать маршруты доступа к данным, сужать и расширять каналы (в том числе динамически).

Одна из ключевых введенных научных гипотез для темы исследований статьи – это обеспечение структурной устойчивости распределенной системы. С этой целью предложено ввести положительные обратные связи и использовать динамические модели в пространстве состояний [17, 18]. Исследования по построению и идентификации управляемых динамических моделей в пространстве состояний описаны в ряде работ [9, 11, 13], однако пока они не получили широкого распространения, так как применялись на аппаратном уровне.

Для реализации принципов проектирования БД необходимо выбрать технологию проектирования систем, обеспечивающую оперативное изменение в ходе экспериментов; реализующую автоматизацию обработки распределенных данных в гибридных хранилищах. Для обеспечения этих функций предложено использовать объектно-реляционные отображения (Object-Relational Mapping, ORM).

Технологии взаимодействия с БД. Развитие структуры взаимодействия информационных систем и приложений с БД привело к появлению таких технологий, как *Open Database Connectivity* (ODBC), *Data Access Object* (DAO), *Borland Database Engine* (BDE), предоставляющих общий программный интерфейс для работы с различными СУБД.

В дальнейшем потребности развития программно-аппаратных средств, работающих с данными, потребовали обеспечения доступа к не SQL хранилищам данных, а к электронной почте и службе каталогов. Для обеспечения этих функций были разработаны технологии *Object Linking and Embedding, Database* (OLEDB) и *ActiveX Data Objects* (ADO). С появлением мощных систем класса Frameworks, таких как *.Net* и *Qt*, технологии обработки данных стали встраиваться в БД, обеспечивая полную интеграцию с ними, а также интеграцию со слабоструктурированными данными в формате XML. Последний стал единым форматом для хранения данных в файлах.

Вычислительные мощности и управляемость каналов передачи данных в гибридной облачной инфраструктуре, открывают широкие возможности по манипуляции с данными как на физическом, так и на логическом уровне, однако, в конечном итоге, приходится оперировать с SQL-запросами, которые в случае распределенных систем имеют перегруженные, громоздкие конструкции.

В условиях широкого распространения объектно-ориентированной методологии разработки прикладных систем и одновременно доминирующего положения на рынке реляционных СУБД, привлекательным решением оказывается использование промежуточного слоя программного обеспечения, предоставляющего необходимые объектно-ориентированные интерфейсы к данным, хранимым под управлением реляционной СУБД [1]. Действительно разработчику намного удобнее оперировать объектами, так как программный код пишется в основном на объектно-ориентированных языках программирования.

Для связи реляционных данных с объектами, с помощью которых разрабатывались программные средства, выбрана технология ORM [7] (рис. 1).

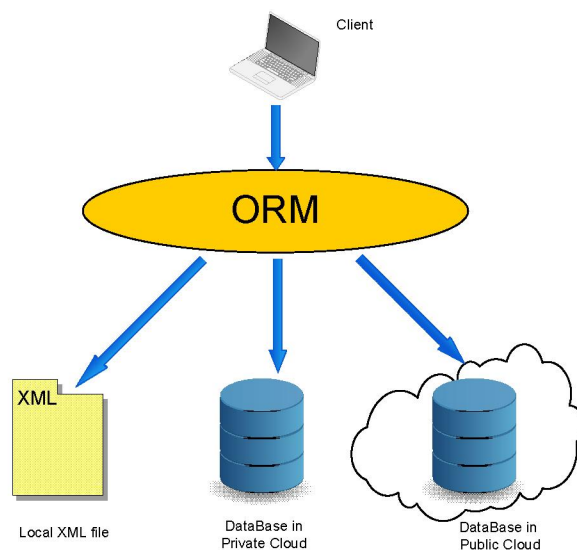


Рис. 1. Взаимодействие клиентских приложений с распределенными хранилищами данных через ORM

Суть данной технологии программирования заключается в обеспечении соответствия сущности реляционной БД объекту. Иными словами, каждому полю таблицы БД ставится в соответствие атрибут класса объекта. Иллюстративный пример такого отражения сущности «студент» изображен на рис. 2.

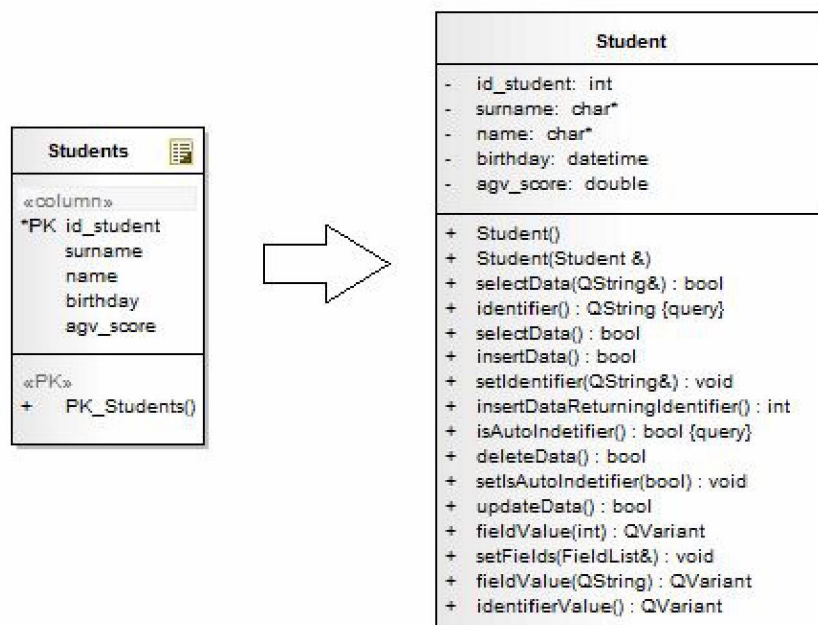


Рис. 2. Пример объектно-реляционного отображения таблицы «Students» в класс «Student»

В этом примере поля таблицы «Students» (*id_student* – уникальный идентификатор студента; *surname* – фамилия студента; *name* – имя студента; *birthday* – дата рождения студента; *agv_sorce* – средний балл студента) отображаются в соответствующие атрибуты класса «Student». После данного отражения в объект включаются методы обработки данных. Таким образом, программисту, использующему объект, нет необходимости строить сложные конструкции SQL, включающие адресацию данных, распределенных в гибридной облачной инфраструктуре. Для этого реализованы методы, которые выполняют операции выборки, вставки, изменения и удаления объекта.

Однако полное отражение реляционных данных в объекты невозможно, поскольку объектная парадигма основана на создании приложений из объектов, имеющих данные и поведение, в то время как реляционная парадигма основана на хранении данных [7].

Проблемы объектно-реляционного отображения распределенных баз данных. Использование промежуточного слоя ORM имеет большое применение в таких реализациях, как *QxORM*, *EntityFramework*, *Dapper*, *Hibernate* и пр. Но все эти технологии эффективно используются лишь для данных, хранящихся и управляющихся только одной СУБД.

Использование технологии ORM позволяет автоматизировать контроль расположения данных. При классическом проектировании разработчик должен обязательно указывать в каждом запросе расположение данных в гибридном облаке, программно подключаться и отключаться от БД. Все это приводит к увеличению трудоемкости разработки программных средств и появлению ошибок в коде. ORM дает возможность включить в каждую сущность атрибут, отвечающий за физическое расположение данных в распределенной системе. В качестве примера на рис. 3 данный атрибут отмечен рамкой.

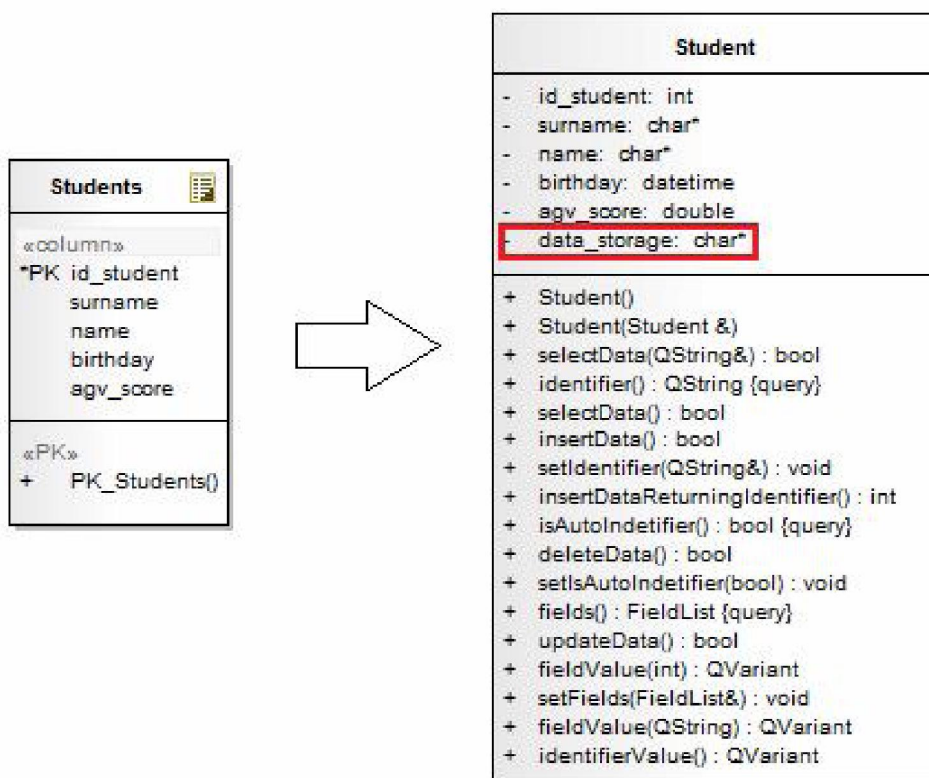


Рис. 3. Пример объектно-реляционного отображения
таблицы «Students» в класс «Student» с атрибутом расположения

При операциях выборки данных в этом атрибуте автоматически устанавливается место хранения данных объекта, а при операциях добавления данный атрибут должен быть установлен программистом. Поэтому актуальной при разработке технологии ORM для гибридной облачной инфраструктуры является разработка интеллектуального модуля, определяющего оптимальное место хранения данных, которое позволит повысить производительность системы. Оптимальность места хранения данных должна автоматически определяться на основе многих критериев, таких как пропускная способность каналов, загруженность серверов, количество клиентов и пр. Многие из этих параметров могут быть получены экспериментальным путем. Поэтому интеллектуальный модуль контроля места хранения данных должен адаптироваться на основе информации, собранной от всех хранилищ данных системы (ЭС) во время опытной эксплуатации.

Другой проблемой при работе с распределенными хранилищами данных является контроль их целостности. При работе с БД, управляемой одной СУБД, за контроль целостности отвечает сама СУБД. При распределенных БД данная функция перекладывается на разработчика программного обеспечения (ПО), так как для повышения надежности ПО данный функционал должен быть специально предусмотрен. Поэтому для автоматизации разработки и увеличения надежности функцию контроля целостности данных в распределенных хранилищах должна выполнять прослойка ORM.

Пример ORM-библиотеки для распределенных БД. Исходя из функций объектно-реляционного отображения данных и особенностей для распределенных баз данных, решающих задачи работы с распределенными БД, можно выделить следующие классы (рис. 4), реализующие ORM библиотеку.



Рис. 4. Структура ORM библиотеки

Классы «DbEntity» обеспечивают, за счет наследования от класса «BasicDbEntity», отражение одной сущности в объект, как показано на рис. 5. Данный класс содержит все необходимые методы для работы с данными, а именно: выборку – методы «SelectData»; добавление – методы «InsertData»; изменение – методы «UpdateData»; удаление – методы «DeleteData». Вся реализация этих методов скрыта от программиста и не требует написания запросов к БД. Единственное, что требуется выполнить, это само отображение, которое происходит в конструкторе класса сущности. В данном примере это класс «Student», в котором описывается соответствие полей таблицы «Students» и атрибутов класса «Student».

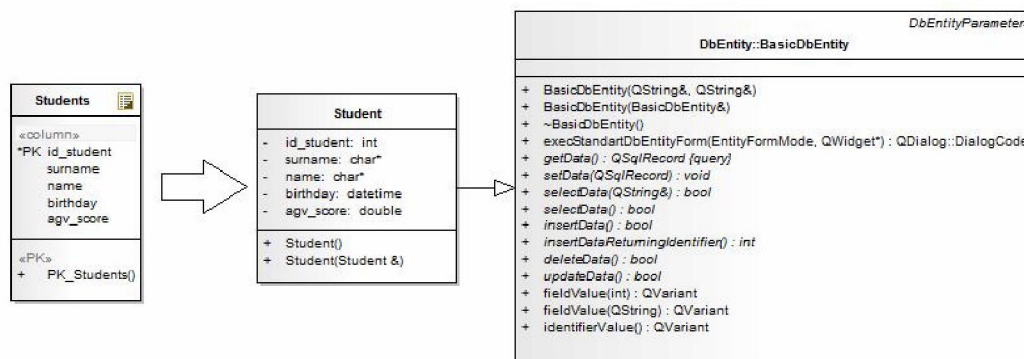


Рис. 5. Пример ORM отображения для сущности «Students» без связей

Классы «DbEntityLink» обеспечивают отражение сущностей со связью «один-ко многим» за счет наследования от «BasicDbEntityLink», как показано на рис. 6. Все действия с данными осуществляются такими же методами, как и в предыдущем случае, потому что класс «BasicDbEntityLink» является наследником класса «BasicDbEntity». Однако при выполнении данных методов уже будут генерироваться другие запросы, обеспечивающие манипулирование также и объектом «Group», который включен в объект «Student». Объект «Group», наследуясь от класса «BasicDbEntity», является отображением сущности «Groups», которая связана с сущностью «Students» связью типа «один-ко многим». Для обеспечения контроля целостности между сущностями, дополнительно при генерации запросов выполняются проверки, блокирующие действия, которые нарушают целостность операции.

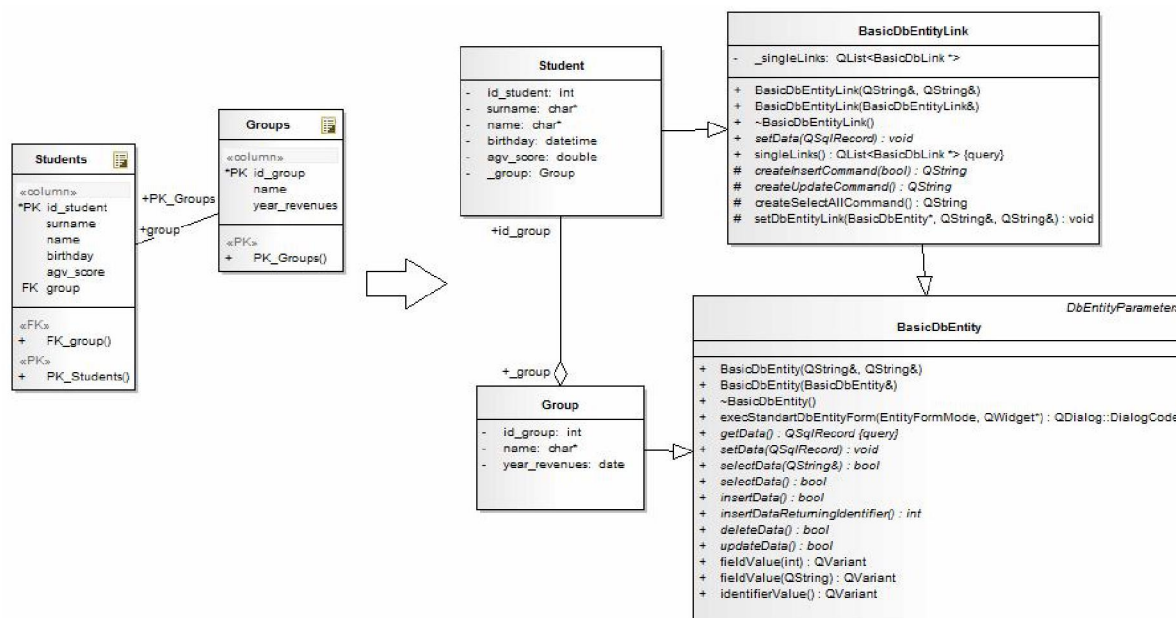


Рис. 6. Пример ORM для сущности «Students» со связью «один-ко многим» с сущностью «Groups»

Классы «DbEntityView» обеспечивают отражение сущностей, связанных со связями «многие-ко многим» и «один-ко многим», за счет наследования от «BasicDbEntityView», как показано на рис. 7.

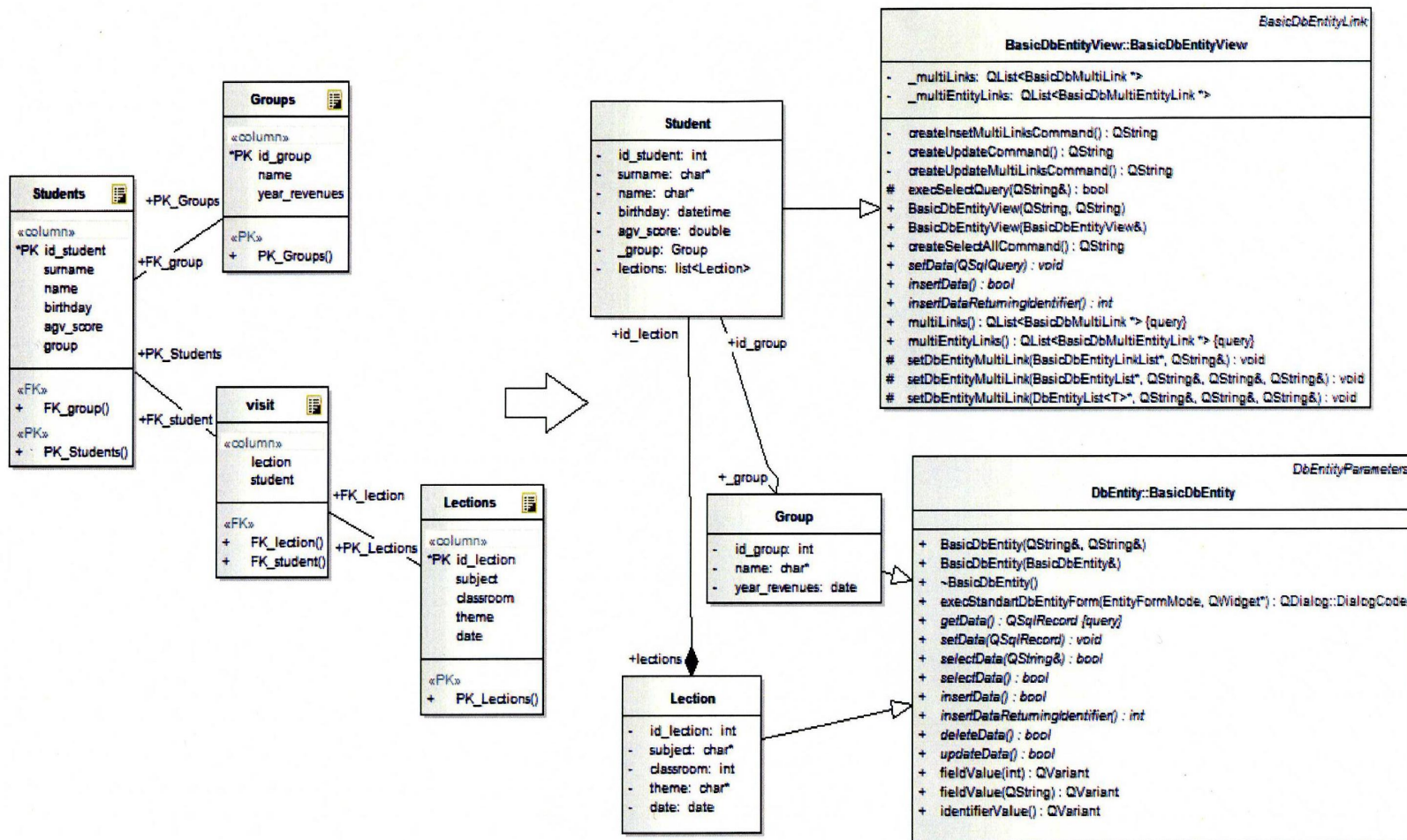


Рис. 7. Пример ORM для сущности «Students» со связями «один-ко многим» с сущностью «Groups» и «многие-ко многим» с сущностью «Lessons»

Класс «BasicDbEntityView», наследуясь от классов «BasicDbEntityLink» и «BasicDbEntity», включает весь их функционал, одновременно добавляя возможность отображения связей «многие-ко многим». Отображение этой связи достигается добавлением контейнера из других объектов. В данном примере добавляется контейнер «list» из объектов «Lecture», которые являются отображением сущности «Lectures», связанной «многие-ко многим» с сущностью «Students».

Классы «DbProcess» обеспечивают подключение к БД и выполнение запросов. Зачастую при разработке с использованием ORM не требуется SQL, так как не исключен случай, когда потребуется выполнить специфичный запрос к БД. Также данные классы используются классами «DbEntity», «DbEntityLink», «DbEntityView» для непосредственного выполнения запросов.

Классы «QueryOptions» обеспечивают хранение запросов в специальной структуре, а также генерацию необходимых запросов выборки с фильтрами и сортировкой. Данные классы используются классами «DbEntity», «DbEntityLink», «DbEntityView» для генерации запросов.

Приведенные на рисунках примеры демонстрируют возможность реализации необходимых функций при проектировании распределенных БД в облачной инфраструктуре.

Выводы. В результате использования технологий программирования объектно-реляционного отображения данных становится возможной реализация контроля расположения и целостности данных, автоматизация разработки информационных систем в гибридной облачной инфраструктуре.

Проектирование ORM-систем на основе использования принципа наследования объектов позволяет вносить изменения для любого из методов объекта без изменения архитектуры системы и полного анализа программного кода. Предложенный подход также дает возможности для реализации обратных связей для программного конфигурирования доступа к данным и обработки запросов. Описанные в статье принципы позволили создать программное обеспечение для ЭС исследования облачно-сетевых технологий.

Список литературы

1. Иванников В. П. Объектно-ориентированное окружение, обеспечивающее доступ к реляционным СУБД / В. П. Иванников, С. С. Гайсарян, К. В. Антипин, В. В. Рубанов // Труды Института системного программирования РАН. – 2001. – Т. 2. – С. 89–114.
2. Курносов М. Г. Децентрализованные алгоритмы диспетчеризации пространственно-распределенных вычислительных систем / М. Г. Курносов, А. А. Пазников // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. – 2012. – № 1. – С. 133–142.
3. Новиков А. М. Разработка технологии и облачной информационной системы для хранения и обработки многомерных массивов научных данных / А. М. Новиков, А. А. Пойда, А. Н. Поляков, С. П. Королев, А. А. Сорокин // Информатика и системы управления. – 2012. – № 4 (34). – С. 156–164.
4. Плужник Е. В. Слабоструктурированные базы данных в гибридной облачной инфраструктуре / Е. В. Плужник, Е. В. Никульчев // Современные проблемы науки и образования. – 2013. – № 4. – С. 95.
5. Плужник Е. В. Лабораторный экспериментальный стенд облачных и сетевых технологий / Е. В. Плужник, Е. В. Никульчев, С. В. Паян // Cloud of Science. – 2014. – Т. 1, № 1. – С. 78–87.
6. Плужник Е. В. Проектирование распределенных систем в гибридной облачной инфраструктуре / Е. В. Плужник, Е. В. Никульчев, С. В. Паян, О. И. Лукьянчиков // Инжиниринг & Телекоммуникации – En&T 2014 : мат-лы междунар. конф. – Москва – Долгопрудный : МФТИ, 2014. – С. 212–214.
7. Ambler S. W. Mapping Objects to Relational Databases: O/R Mapping In Detail / S. W. Ambler. – Ambrysoft Inc., 2010.
8. Buyya R. Autonomic cloud computing: Open challenges and architectural elements / R. Buyya, R. N. Calheiros, X. Li // 2012 Third International Conference on Emerging Applications of Information Technology (EAIT). – IEEE Publ., 2012. – P. 3–10.

9. Hellerstein J. L. Challenges in Control Engineering of Computing Systems. IBM Research Report. RC23159 (W0309-091) / J. L. Hellerstein. – NY : Thomas J. Watson Research Center, 2003.
10. Jurczyk P. Dynamic Query Processing for 2p Data Services in the Cloud / P. Jurczyk, L. Xiong // Database and Expert Systems Applications. – Berlin – Heidelberg, 2009. – P. 396–411.
11. Lemmon M. D. Towards a passivity framework for power control and response time management in cloud computing / M. D. Lemmon // Proceedings of 7th International Workshop on Feedback Computing. – 2012.
12. Mahmood Z. (ed.) Cloud Computing. Challenges, Limitations and R&D Solutions / Z. Mahmood– Springer International Publishing, 2014.
13. Nathuji R. Q-clouds: managing performance interference effects for QoS-aware clouds / R. Nathuji, A. Kansal, A. Ghaffarkhan // Proc. of the 5th European conference on Computer systems. – ACM, 2010. – P. 237–250.
14. Pandey S. An autonomic cloud environment for hosting ECG data analysis services / S. Pandey, W. Voorsluys, S. Niu, A. Khandoker, R. Buyya // Future Generation Computer Systems. – 2012. – Vol. 28, № 1. – P. 147–154.
15. Pluzhnik E. V. Use of Dynamical Systems Modeling to Hybrid Cloud Database / E. V. Pluzhnik, E. V. Nikulchev // Int'l J. of Communications, Network and System Sciences. – 2013. – Vol. 6, № 12. – P. 505–512.
16. Pluzhnik E. Virtual laboratories in cloud infrastructure of educational institutions / E. Pluzhnik, E. Nikulchev // 2nd International Conference on Emission Electronics (ICEE). Selected papers. – IEEE; 2014. – P. 67–69.
17. Pluzhnik E. Concept of Feedback in Future Computing Models to Cloud Systems / E. Pluzhnik, E. Nikulchev, S. Payain // World Applied Sciences Journal. – 2014. – Vol. 32, № 7. – P. 1394–1399.
18. Pluzhnik E. Optimal control of applications for hybrid cloud services / E. Pluzhnik, E. Nikulchev, S. Payain // Proc. 2014 IEEE 10th World Congress on Services (SERVICES 2014). Anchorage, USA. – IEEE, 2014. – P. 458–461.
19. Sithole E. Cache Performance Models for Quality of Service Compliance in Storage Clouds / E. Sithole, A. McConnell et al. // Journal of Cloud Computing: Advances, Systems and Applications. – 2013. – Vol. 2, № 1. – P. 1–24.

References

1. Ivannikov V. P., Gajjarjan S. S., Antipin K. V., Rubanov V. V. Obektno-orientirovannoe okruzhenie, obespechivayushchee dostup k relyatsionnym SUBD [Object-oriented environment that provides access to relational DBMS]. *Trudy Instituta sistemnogo programmirovaniya RAN* [Proceedings of Institute for System Programming], 2001, vol. 2, pp. 89–114.
2. Kurnosov M. G., Paznikov A. A. Decentralizovannyye algoritmy dispatcherizatsii prostranstvenno-raspredelejnykh vychislitelnykh sistem [Decentralized algorithms of dispatching of distributed computing systems]. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitel'naya tekhnika i informatika* [Bulletin of Tomsk State University. Management, computer science and informatics], 2012, no. 1, pp. 133–142.
3. Novikov A. M., Poyda A. A., Polyakov A. N., Korolev S. P., Sorokin A. A. Razrabotka tekhnologii i oblachnoy informatsionnoy sistemy dlya khraneniya i obrabotki mnogomernykh massivov nauchnykh dannykh [Development of technology and cloud information system for the storage and processing of multidimensional scientific datasets]. *Informatika i sistemy upravleniya* [Informatics and Management Systems], 2012, no. 4 (34), pp. 156–164.
4. Pluzhnik E. V., Nikulchev E. V. Slabostrukturirovannyye bazy dannykh v gibridnoy oblachnoy infrastrukture [Semistructured database in a hybrid cloud infrastructure]. *Sovremennyye problemy nauki i obrazovaniya* [Modern Problems of Science and Education], 2013, no. 4, p. 95. (In Russ.)
5. Pluzhnik E. V., Nikulchev E. V., Payain S. V. Laboratornyy eksperimentalnyy stend oblachnykh i setevykh tekhnologiy [Laboratory test bench cloud and network technologies]. *Cloud of Science*, 2014, vol. 1, no. 1, pp. 78–87.
6. Pluzhnik E. V., Nikulchev E. V., Payain S. V., Lukyanchikov O. I. Proektirovanie raspredelennykh sistem v gibridnoy oblachnoy infrastrukture [Design of distributed systems in a hybrid cloud infrastructure]. *Inzhiniring & Telekommunikatsii – En&T 2014: materialy mezhdunarodnogo konferentsii «Engineer-*

ing & Telecommunications – En&T 2014: Proceedings of the International Conference. Moscow, Dolgoprudny, 2014, pp. 212–214.

7. Ambler S. W. Mapping Objects to Relational Databases: O/R Mapping In Detail. Ambysoft Inc., 2010.

8. Buyya R., Calheiros R. N., Li X. Autonomic cloud computing: Open challenges and architectural elements. *2012 Third International Conference on Emerging Applications of Information Technology (EAIT)*, 2012, pp. 3–10.

9. Hellerstein J. L. Challenges in Control Engineering of Computing Systems. IBM Research Report. RC23159 (W0309-091). NY, 2003

10. Jurczyk P., Xiong L. Dynamic Query Processing for 2p Data Services in the Cloud. *Database and Expert Systems Applications*, Berlin, Heidelberg, 2009, pp. 396–411.

11. Lemmon M. D. Towards a passivity framework for power control and response time management in cloud computing. *Proceedings of 7th International Workshop on Feedback Computing*, 2012.

12. Mahmood Z. (ed.) Cloud Computing. Challenges, Limitations and R&D Solutions. Springer International Publishing, 2014.

13. Nathuji R., Kansal A., Ghaffarkhah A. Q-clouds: managing performance interference effects for QoS-aware clouds. *Proc. of the 5th European conference on Computer system*, 2010, pp. 237–250.

14. Pandey S., Voorsluys W., Niu S., Khandoker A., Buyya R. An autonomic cloud environment for hosting ECG data analysis services. *Future Generation Computer Systems*, 2012, vol. 28, no. 1, pp. 147–154.

15. Pluzhnik E. V., Nikulchev E. V. Use of Dynamical Systems Modeling to Hybrid Cloud Database. *Int'l J. of Communications, Network and System Sciences*, 2013, vol. 6, no. 12, pp. 505–512.

16. Pluzhnik E., Nikulchev E. Virtual laboratories in cloud infrastructure of educational institutions. *2nd International Conference on Emission Electronics (ICEE). Selected papers*, 2014, pp. 67–69.

17. Pluzhnik E., Nikulchev E., Payain S. Concept of Feedback in Future Computing Models to Cloud Systems. *World Applied Sciences Journal*, 2014, vol. 32, no. 7, pp. 1394–1399.

18. Pluzhnik E., Nikulchev E., Payain S. Optimal control of applications for hybrid cloud services. *Proc. 2014 IEEE 10th World Congress on Services (SERVICES 2014). Anchorage, USA*, 2014, pp. 458–461.

19. Sithole E., McConnell A. et al. Cache Performance Models for Quality of Service Compliance in Storage Clouds. *Journal of Cloud Computing: Advances, Systems and Applications*, 2013, vol. 2, no. 1, pp. 1–24.

УДК 004.2

ИНТЕГРАЛ ШОКЕ В ТЕОРИИ ЭКОНОМИКИ АКТИВНОГО КОННЕКТА

Статья поступила в редакцию 23.10.2014, в окончательном варианте 11.12.2014.

Лавриченко Олег Вячеславович, кандидат экономических наук, инженер-программист, ОАО «Концерн Моринформсистема-Агат», 111024, Российская Федерация, г. Москва, шоссе Энтузиастов, 29, e-mail: lavrslava1962@mail.ru

Показана актуальность использования инновационных самоорганизующихся бизнесобразующих технологий в современных условиях экономического развития России. Рассмотрены основы авторской теории экономики активного коннекта. Обоснована необходимость снижения действия антропогенных факторов на эффективность использования аддитивных моделей и алгоритмов при принятии решений. Показана необходимость эконормирования антропогенных сукцессий в рамках концепции сбалансированного распределения инновационных ресурсов промышленных предприятий между объектами инноваций. Исследованы существующие подходы к задаче оптимизации интеграла Шоке и возможности его применения при принятии решений, направленных на устранение диспаритета в распределении инновационных ресурсов между объектами инноваций. Обосновано, что применение интеграла Шоке существенно расширяет возможности и эффективность оптимизации решений для многокритериальных задач распределения инновационных ресурсов в условиях неопределенности. Доказана необходимость ослабления независимости бинарных отношений объектов инноваций и