

Разработка программного приложения для синхронизации базы данных справочника материалов и сортаментов

Д.Е. Баранов, В.А. Давыденко, А.В. Цупранков, А.Г. Янишевская, В.Б. Василевский, В.В. Соседко

В настоящее время проблема синхронизации электронных данных для поддержания их в актуальном и достоверном состоянии в различных подразделениях предприятий очень актуальна, особенно в условиях отсутствия возможности проведения локальной вычислительной сети вследствие их удаленного расположения друг от друга [1, 2]. Частично проблему можно решить с помощью внешних носителей информации путем копирования и доставки между отделами. Однако здесь возникает другая проблема: как быть с более объёмными и динамически изменяющимися структурами данных – базами данных. Простое копирование с одной вычислительной машины на другую проблему не решает, так как будут сохранены лишь изменения, внесенные только в одной из баз данных, изменения же второй базы будут утеряны либо неизбежно возникнет дублирование части данных [3].

Целью работы являлась разработка программного приложения, позволяющего сравнивать содержимое двух баз данных.

Задача работы заключалась в автоматизированном добавлении новых элементов в базу данных корпоративного справочника материалов и сортаментов компании АСКОН [4], расположенную в любом из подразделений предприятия, из баз данных других подразделений, для возможной работы с ними, причем базы данных в подразделениях имеют полностью идентичные структуры. При этом имеется возможность переносить копии баз данных между подразделениями на внешнем носителе информации.

Актуальность работы в том, что база данных справочника достаточно объёмная, может пополняться новыми элементами различными пользователями сразу, вследствие чего исключается какая либо ручная обработка и контроль новых элементов. Разработанное приложение должно позволить в автоматизированном порядке находить различия двух баз данных, дополнять базу данных любого из подразделений элементами, которые отсутствуют в ней и имеются в базах данных других подразделений. Это упростит процесс синхронизации данных о материалах и сортаментах в подразделениях предприятия и обеспечит единую работу с ними.

Выполнение задачи проводилось в несколько этапов:

- изучение структуры базы данных;
- выбор алгоритма синхронизации баз данных;
- выбор среды разработки и кодирование приложения;
- отладка и тестирование приложения в реальных условиях.

На первом этапе была изучена структура базы данных справочника.

Анализ проводился в два подэтапа:

- нахождение первичных ключей;
- определение таблиц, требуемых выполнения синхронизации.

Первичный ключ представляет собой один из примеров уникальных индексов и применяется для уникальной идентификации записей таблицы. Никакие из двух записей таблицы не могут иметь одинаковых значений первичного ключа. Другими словами первичный ключ это идентификационный номер, уникальный для каждого материала или изделия [5, 6].

В таблицах BOLD_OBJECT и BOLD_TYPE другой первичный ключ – BOLD_TYPE. В данном случае это пример реализации отношений «один к множеству», который связывает идентификационные номера с их типами. Типы нужны для сортировки в справочнике материалов и сортаментов АСКОН объектов по различным категориям. Так как перечень типов не изменяется со временем работы программы, BOLD_TYPE не является

необходимым ключом для синхронизации. Поле BOLD_ID присутствует в нескольких таблицах – это первичный ключ в этих таблицах. Это идентификатор многих основных объектов в базе данных программы (материалов, изделий, дополнительных свойств и других). Все таблицы содержащие BOLD_ID необходимо синхронизировать.

Вторым этапом был выбор возможных путей анализа и сравнения элементов баз данных. По первичному ключу BOLD_ID из таблиц выбирались параметры материалов и сортовентов, по которым было возможно сопоставление для выявления различий в двух материалах или сортаментах. На их основе строились временные объекты, собираемые в два динамических массива (по одному для каждой из сравниваемых баз данных). Эти массивы и подлежали дальнейшему по-объектному сравнению. Объекты с равными параметрами, имеющиеся в обоих массивах удалялись из них. В результате сравнения оставались массивы, содержащие только различающиеся объекты. Необходимым было пополнение новыми элементами только первой базы данных (производится пополнение базы данных одного подразделения из копии базы данных другого подразделения). Следовательно, по первичному ключу объектов, оставшихся во втором массиве, находились записи материалов и сортовентов во второй базе данных и добавлялись в первую (включая параметры, не используемые для создания объекта сравнения).

На третьем этапе языком программирования был выбран язык C++, для реализации программы использовалось объектно-ориентированное программирование, а средой разработки – C++Builder производства корпорации Borland, предназначенный для операционных систем платформы Win32. C++Builder – это законченный, гибкий инструмент RAD, сочетающий мощь современного компилятора языка C++, удобство средств визуальной разработки приложений, достоинства компонентного подхода к проектированию с открытыми возможностями пополнения имеющихся в распоряжении разработчика компонент [7, 8].

Поскольку данное приложение разрабатывается для работы в среде операционной системы Windows, то для его реализации был использован стандартный подход при написании приложения с использованием графического интерфейса. Была создана форма с визуальными компонентами, для которых были сопоставлены функции-обработчики. После того как происходит событие на каком-то из визуальных компонент, (щелчок мыши, нажатие на кнопку или пункт меню), операционная система посылает приложению соответствующее сообщение, и запускается соответствующий обработчик именно для этого события. При разработке приложения было выделено два подэтапа:

1) Разработка визуального интерфейса пользователя, на основе экранных форм среды C++ Builder, визуальных и невидимых компонентов, предназначенных для подключения к исходным данным, запуска действий по обработке исходных данных, вывода результата обработки, вывода результатов на экранную форму [9].

2) Разработка обработчиков событий от элементов экранных форм, разработка программных кодов действий обработки содержимого баз данных [10].

Интерфейс был разработан таким образом, чтобы созданное приложение автоматически подключалось к базам данных, при этом пользователем заранее должна быть восстановлена копия второй базы данных, с которой, собственно, и будет сравниваться исходная база данных.

На следующем шаге обработчики событий от объектов экранных форм были реализованы как функции. Функции в свою очередь могут вызывать функции более низкого уровня и так далее. Таким образом, программа имеет иерархическую структуру.

В разработанном приложении содержится описание 7 функций обработчиков событий:

- `datasynch` – сравнение текущей даты и дат последних синхронизации и изменений баз данных;

- readobject – чтение новых объектов (материалов или сортов) с базы данных;
- bdsort – поиск отличий, описание структур;
- SpeedButton1Click – запуск поиска изменений;
- SpeedButton2Click – сохранение найденных объектов (если таковые были);
- FormCreate – установка формы по центру;
- FormShow – определение даты.

Заключительным этапом была разработка самого приложения и его отладка. При отладке приложения нужно было проверить соответствие разработанного приложения исходному заданию, проверить его на отсутствие ошибок, провести несколько тестовых сравнений с заранее подготовленными наборами баз данных с небольшим количеством элементов.

Главное меню содержит пункты (см. рис. 1):

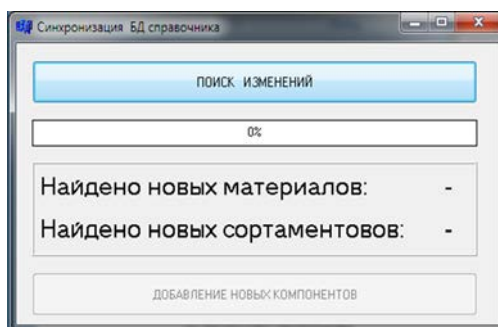


Рис. 1 – Главное меню программы

- «ПОИСК ИЗМЕНЕНИЙ» (при нажатии выполняется поиск отличий объектов в базах данных);
- «Найдено новых материалов» (отображает количество материалов, добавленных только во вторую базу данных);
- «Найдено новых сортов» (отображает количество сортов, добавленных только во вторую базу данных);
- «ДОБАВЛЕНИЕ НОВЫХ КОМПОНЕНТОВ» (активируется при наличии компонентов, добавленных только во вторую базу данных, при нажатии добавляет найденные компоненты в первую базу данных).

Для поиска и добавления новых элементов в базу данных справочника на одном из подразделений предприятия необходимо:

- на сервере в отдельную базу данных восстановить резервную копию, созданную из базы данных справочника;

- запустить исполняемый файл приложения «BDSort.exe», при появлении диалогового окна с указанной ошибкой устранить её и перезапустить приложение;

- запустить работу приложения нажатием кнопки «ПОИСК ИЗМЕНЕНИЙ» (см. рис. 2);

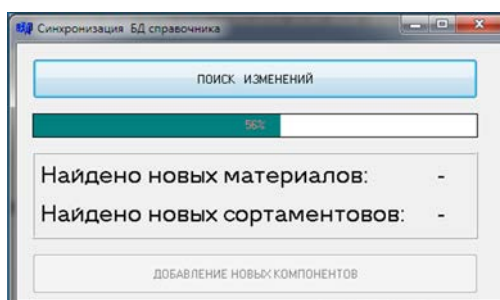


Рис. 2 – Ход работы программы

- информация об успешном завершении работы отобразится в диалоговом окне (см. рис. 3);

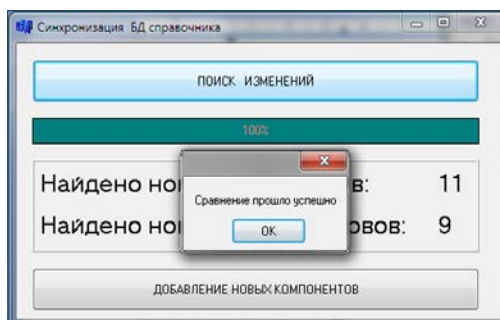


Рис. 3 – Результат работы программы

- в главном меню программы будет указано количество новых материалов и сортаментов;

- нажатием кнопки «ДОБАВЛЕНИЕ НОВЫХ КОМПОНЕНТОВ» они будут внесены в базу данных.

Разработанное приложение было апробировано и внедрено на одном из омских предприятий. Оно также может быть модернизировано для работы в

автоматическом режиме и подключения к удалённым базам данных, при объединении подразделений предприятия в локальную вычислительную сеть.

Литература:

1. Галушка, В.В. Формирование обучающей выборки при использовании искусственных нейронных сетей в задачах поиска ошибок баз данных [Электронный ресурс] / В.В. Галушка, В.А. Фатхи // Инженерный вестник дона. Электронный журнал. - 2013. - № 2. - Режим доступа: <http://ivdon.ru/magazine/archive/n2y2013/1597>.

2. Jagadish, H.V. Integrity Maintenance in an Object-Oriented Database / H.V. Jagadish, Qian Xiaolei - AT&T Bell Laboratories Murray Hill, New Jersey 07974. Proceedings of the 18th VLDB Conference Vancouver, British Columbia, Canada, 1992.

3. Целигорова, Е.Н. Современные информационные технологии и их использование для исследования систем автоматического управления [Электронный ресурс] / Е.Н. Целигорова // Инженерный вестник дона. Электронный журнал. - 2010. - № 3. - Режим доступа: <http://ivdon.ru/magazine/archive/n3y2010/222>.

4. Корпоративный справочник Материалы и Сортаменты 2.1 - Электронное справочное руководство. - ЗАО Аскон, 2009.

5. Хомоненко, А. Д. Базы данных / А.Д. Хомоненко, В.М. Цыганков, М.Г. Мальцев – Санкт -Петербург: БХВ-Петербург, 2004. – 747 с.: ил.

6. Wiederhold, G. Views, Objects, and Databases / G. Wiederhold - IEEE Computer, 19 (12), December 1986. 37-44.

7. Архангельский, А.Я. Программирование в С++ Builder 6 и 2006 / А.Я. Архангельский, М.А. Тагин – Москва: ООО «Бином-Пресс», 2007. – 1184 с.: ил.

8. Холингворт, Дж. Borland С++Builder 6. Руководство разработчика. / Дж. Холингворт, Б. Сворт, М. Кэшмэн, П. Густавсон П.: Пер. с англ. - Москва: Издательский дом «Вильяме», 2004. – 976 с.: ил.

9. Волоха, А.Н. Microsoft SQL Server 2005. Новые возможности. Санкт - Петербург: Издательство Питер, 2006. – 304 с.: ил.

10. Хомоненко, А.Д. Работа с базами данных в С++ Builder / А.Д. Хомоненко, С.Е. Ададуров. – Санкт-Петербург: БХВ-Петербург, 2006. – 496 с.: ил.