

## МЕТОДЫ УПРАВЛЕНИЯ СОПРОВОЖДЕНИЕМ СЛОЖНЫХ ИНФОРМАЦИОННЫХ СИСТЕМ С ИЕРАРХИЧЕСКОЙ СТРУКТУРОЙ

Орлов В.Л., Курако Е.А.

*Институт проблем управления им. В.А. Трапезникова РАН,*

*Россия, г. Москва, ул. Профсоюзная д.65*

*ovl@ipu.ru, kea@ipu.ru*

*Аннотация: Рассмотрены методы управления сопровождением сложных информационных систем с иерархической структурой. Предложен сценарный подход к организации обновления в таких системах. Определены основные методы обновления серверов приложений, серверов баз данных и клиентских мест в процессе сопровождения.*

Ключевые слова: сопровождение; обновление программ; сложные информационные системы; сервер приложений; база данных; клиент.

### **Введение**

Задача управления сопровождением сложных информационных систем в последнее время приобретает все большую актуальность. Это определяется следующими моментами:

- сложностью архитектуры систем, которые содержат несколько уровней, причем уровни могут быть территориально распределены;
- разнородностью внутриуровневого построения, которое предусматривает как минимум три принципиально различных базовых компонента: клиентское место, сервер приложений и сервер баз данных (БД);

- необходимостью синхронизации изменений не только внутри уровня, но и между уровнями;
- возможностью только кратковременных остановок, предназначенных для изменения программного обеспечения, или (что желательно) отсутствие таковых.

Естественно, что наличие всех этих условий делает задачу нетривиальной. В то же время решение ее необходимо, так как практика использования программ любой сложности показывает, что их модернизация в процессе эксплуатации является востребованной. Это позволяет не только устранять ошибки, но и вносить новые функциональные возможности.

### 1 Сценарный подход в процессе подготовки коррекции программ и данных

Как отмечалось выше, на каждом уровне системы могут быть выделены такие компоненты, как клиентские места, сервера приложений и сервера БД (рис.1).

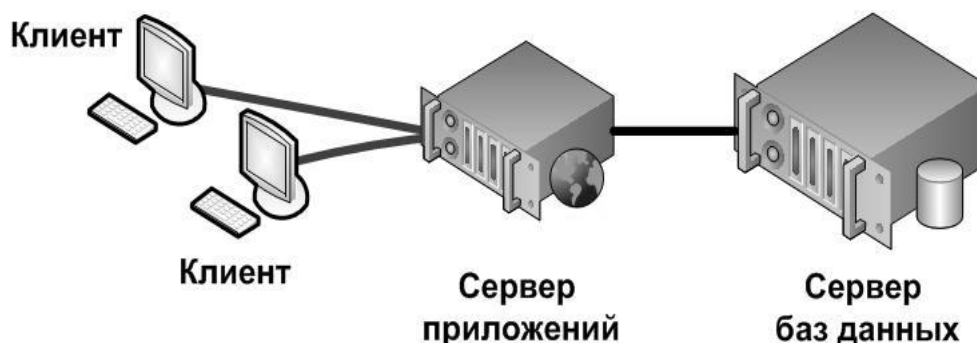


Рис. 1. Компоненты одного уровня

Из всего многообразия объектов, представленных на компонентах, целесообразно выделить следующие [1]:

- На клиенте:
  - Программные модули и библиотеки;
  - Директории и файлы.
- На сервере приложений:
  - Программные модули и библиотеки;
  - Директории и файлы;
  - Службы операционной системы;
  - Электронные сервисы.
- На сервере баз данных:
  - Хранимые процедуры;
  - Структура БД (таблицы, представления, последовательности, связи и т.д.).

Обработка объектов разных типов ведется различными способами. Основные типы обработки представлены в таблице 1.

Таблица 1. Виды обработки объектов

Объекты	Виды обработки
Директории	Создание, копирование, удаление
Файлы	Копирование, удаление
Модули	Копирование, удаление
Библиотеки	Копирование, удаление
Службы и сервисы	Регистрация, копирование, удаление
Хранимые процедуры БД	Создание, замена, удаление
Структура БД	Создание, модификация, удаление
Управляющая информация системы, хранящаяся в БД	Создание, модификация, удаление

Как мы видим, обработка различных типов объектов проводится по-разному. В то же время целесообразно проводить группировку обработки нескольких объектов или даже нескольких типов объектов в одном пакете. Это объясняется тем, что, например, при добавлении какой-либо новой функции требуется изменение как программ клиента, так и серверных программ, а также необходимо проведение коррекции БД. Все эти действия в необходимой последовательности могут быть корректно выполнены, если их реализация будет определяться сценарием. Поэтому вместе с

подготовкой пакета коррекции необходимо указывать тип объекта, вид обработки, а также место хранения информации и место, где ее требуется разместить. Все эти данные попадают в автоматически формируемый сценарий.

То есть в итоге пакет включает в свой состав сценарий, а также все необходимые объекты, описания действий с которыми имеется в указанном сценарии.

Следует отметить, что объекты БД должны включаться в пакет в виде скриптов, описывающих их создание и изменения.

## **2 Алгоритм обработки сценария на серверах**

Важным элементом сценария является номер версии пакета. Он используется при проверке необходимости обновления. Все подготовленные пакеты помещаются на центральный сервер обновлений (ЦСО). В сценарии каждого пакета содержится номер его версии.

Все периферийные службы обновления (ПСО) периодически обращаются к ЦСО, который предоставляет им список хранящихся у него пакетов, содержащих обновления. В списке у каждого пакета присутствует номер его версии.

После получения списка ПСО сверяет список с имеющимся у нее списком установленных обновлений, выбирает первое неустановленное обновление, запрашивает его у ЦСО и производит установку. Здесь важным является то обстоятельство, что список задает последовательность установки обновлений. Это существенно, так как нарушение этой последовательности может привести к невозможности инсталляции того или иного пакета. То есть здесь работает метод исключения пропуска установки. Пока не установится определенное обновление, последующие также не будут устанавливаться.

Например, если подготовленный пакет по каким-либо причинам содержит ошибку, которая не дает возможности его установить, то запросы от ПСО будут периодически повторяться, до тех пор, пока в ЦСО не появится новая корректная версия пакета.

Если же вновь установленный пакет не удовлетворяет сотрудников, обеспечивающих эксплуатацию, то возможен возврат на предыдущую версию по команде оператора ПСО. В этом случае для обновления будет приниматься только следующая версия, номер которой превышает номер версии, забракованной на этом сервере.

## **3 Обновление сервера приложений**

На сервере приложений обновляются файлы, библиотеки, директории, электронные сервисы и службы, связанные с операционной системой. Если файлы замещаются простым копированием в соответствии с параметрами, указанными в сценарии, то электронные сервисы и службы, необходимо, как правило, регистрировать. Эти дополнительные действия определяются типом объекта, также указанным в сценарии. Отметим, что при обработке объекта «директория» иногда необходимо создавать пустые директории. В таком случае используется не процедура копирования, а команда создания директории.

## **4 Обновление базы данных**

База данных не обновляется копированием. Здесь для изменения структуры базы данных, что часто является необходимым действием при сопровождении, и для изменения хранимых процедур используются SQL-скрипты, выполняемые службой обновления на сервере приложений в соответствии со сценарием и корректирующие связанную базу данных. То есть при обновлении баз данных фактически используются вложенные сценарии.

Важным обстоятельством является то, что при подготовке новых модулей и библиотек для клиента их необходимо зарегистрировать во фрагменте БД, содержащим управляющую информацию системы. Такая регистрация проводится службой обновления на сервере приложений.

## **5 Обновление клиента**

Программное обеспечение клиента обладает двумя особенностями. Во-первых, клиентские программы взаимодействуют с БД только через сервисы. Следовательно, они ничего не знают о структуре БД и принципиально не могут выполнять никакие SQL-скрипты. Во-вторых, программное обеспечение клиента не включает в свой состав электронные сервисы и службы. То есть здесь не требуется проводить процедуры регистрации, связанные с операционной системой. Из таблицы 1 следует, что в этом случае для обновления программного обеспечения достаточно использовать простые процедуры создания, копирования и удаления.

Таким образом, для выполнения функций обновления клиента служба обновления размещает в заданном каталоге сервера приложений обновляемые модули, библиотеки, файлы и директории клиента (обобщенные файлы). Управляющая программа клиента в определенные моменты времени,

например, при перезапуске прикладной программы клиента производит сравнение имеющихся в каталоге обобщенных файлов на сервере с аналогичными файлами клиента. Все модули, библиотеки, файлы и директории, которые были обновлены на сервере, копируются в соответствующие каталоги клиента и автоматически подхватываются при старте.

### **Заключение**

Одним из важных элементов процесса сопровождения сложных программных комплексов является обновление программ, структуры данных и данных на всех уровнях. В связи с тем, что в этом случае на всех уровнях используются разнообразные процедуры, определяемые типом объектов обновления, целесообразно применение сценарного подхода. Применение вложенных сценариев, обеспечивает выполнение процедур обновления для всех основных типов объектов.

### **Литература**

1. *Курако Е.А., Орлов В.Л.* Методы динамического обновления программных комплексов для систем с каскадной структурой // Проблемы управления. М.,2017. № 5. С. 52-60.