

СПОСОБЫ ВЗАИМОДЕЙСТВИЯ С УДАЛЕННЫМ РЕПОЗИТОРИЕМ ЯЗЫКА ПИФАГОР

Ермаков В.Н.,

научный руководитель д-р техн. наук Легалов А.И.

Сибирский федеральный университет

Для современных интегрированных сред разработки правилом хорошего тона уже давно является поддержка систем управления версиями (СУВ), не стала исключением и среда разработки функционально-потокowego языка Пифагор для создания архитектурно-независимых параллельных программ [1]. Вместе с тем, помимо базовых функциональных возможностей классического удаленного репозитория, данный вариант включает в себя функции класса программ, называемых хранилищем данных. Программных подсистем, сочетающих в себе функции СУВ, поисковой машины и системы управления базами данных (СУБД). В функционал хранилища данных входит:

- хранение структурированных и неструктурированных данных,
- полнотекстовый поиск,
- версионирование,
- поддержка транзакций,
- наблюдение за контентом.

Обобщенный вид структурной схемы удаленного репозитория может быть пояснен с помощью следующего рисунка:

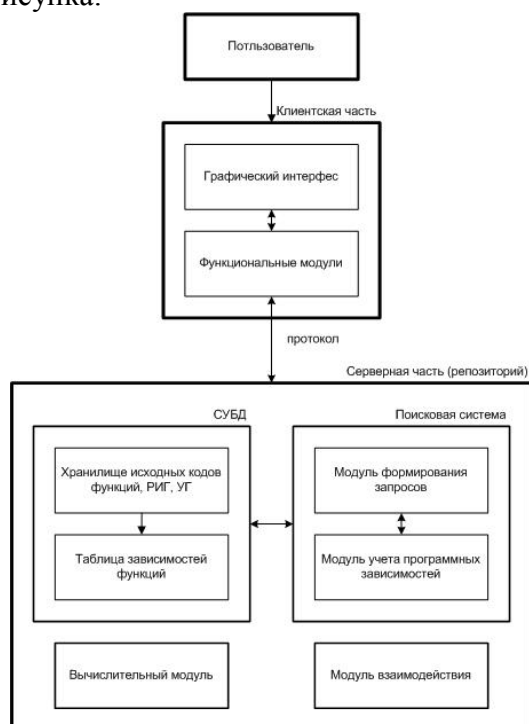


Рисунок 1. Структурная схема репозитория.

Процесс взаимодействия пользователя с данным репозиторием через графический интерфейс интегрированной среды разработки может быть отражен с помощью следующих пунктов:

- пользователь запускает среду разработки;
- с помощью графического интерфейса пишет программу или некоторую функцию;

- при появлении необходимости сохранить написанный исходный код программы или функции он выбирает пункт меню "Файл\Сохранить в репозитории" или же нажимает кнопку "Сохранить в репозитории".
- при этом, если ранее в настройках программы уже был указан репозиторий, то программа попытается подключиться к нему, используя сетевой протокол HTTP, HTTPS, FTP,.. И выполнить операцию записи новой функции или же группы функций в СУБД, при этом она/они будут проанализированы на предмет наличия взаимосвязей, а результаты анализа будут записаны в таблицу зависимости функций. Формат записи системного имени функций будет иметь иерархическую структуру, благодаря использованию в нем разделителя "." (например, "math.user1.factorial.v1"), позволяющего гибко задавать структуру пространств имен и их подпространств. В случае отсутствия соответствующих настроек на стороне клиента, пользователю будет предложено указать их.
- если же пользователю необходимо воспользоваться в своей программе написанной ранее функцией или процедурой, то он выбирает пункт меню "Файл\Добавить библиотеку" или кнопку "Добавить библиотеку", при этом ему открывается диалоговое окно вида:

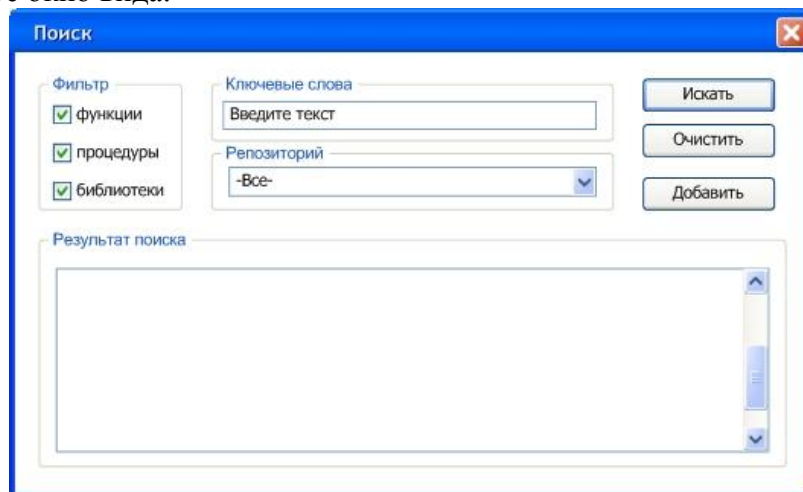


Рисунок 2. Окно поиска.

После заполнения необходимых полей, пользователь нажимает кнопку "Искать", отправляя на выбранные репозитории запрос на поиск соответствующих элементов. Модуль взаимодействия получает запрос и поисковая система формирует запрос к СУБД, которая в свою очередь формирует на него ответ, учитывая при этом разграничения прав доступа, как на отдельные элементы, так и на весь репозиторий в целом. Результаты работы соответствующих репозиториях асинхронно отправляются на клиент и отображаются в поле диалогового окна поиска. Пользователь выделяет строку с записью об искомом элементе и нажимает кнопку "Добавить". При этом на соответствующий репозиторий отправляется запрос о получении данного элемента. Поисковая система находит в СУБД ее исходный код и анализирует таблицу зависимостей на предмет наличия необходимости копирования дополнительных функций из данного или удаленного репозитория, если такая необходимость имеет место быть. В последнем случае, поисковая система с помощью модуля взаимодействия обращается к репозиторию, на который ссылается функция. А тот в свою очередь повторяет операции предыдущего. При этом на каждом из узлов формируется пакет данных с перечнем необходимых исходных кодов, их реверсивных информационных (РИГ) и

управляющих графов (УГ), которые в конечном итоге оказываются аккумулированы в полном объеме на рабочей станции пользователя (Рисунок 3).

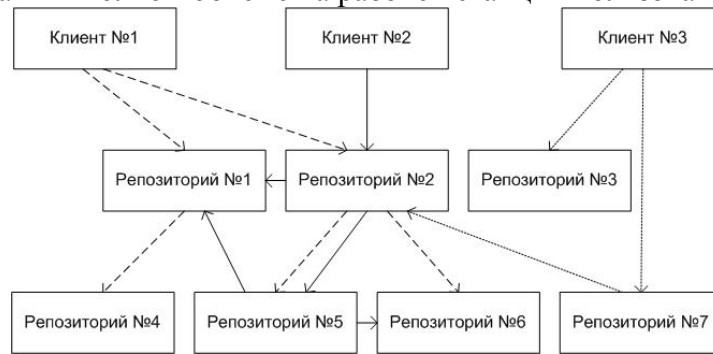


Рисунок 3. Пример.

Программный интерфейс клиентской части системы удаленного репозитория может быть отражен с помощью следующей диаграммы классов (Рисунок 4).

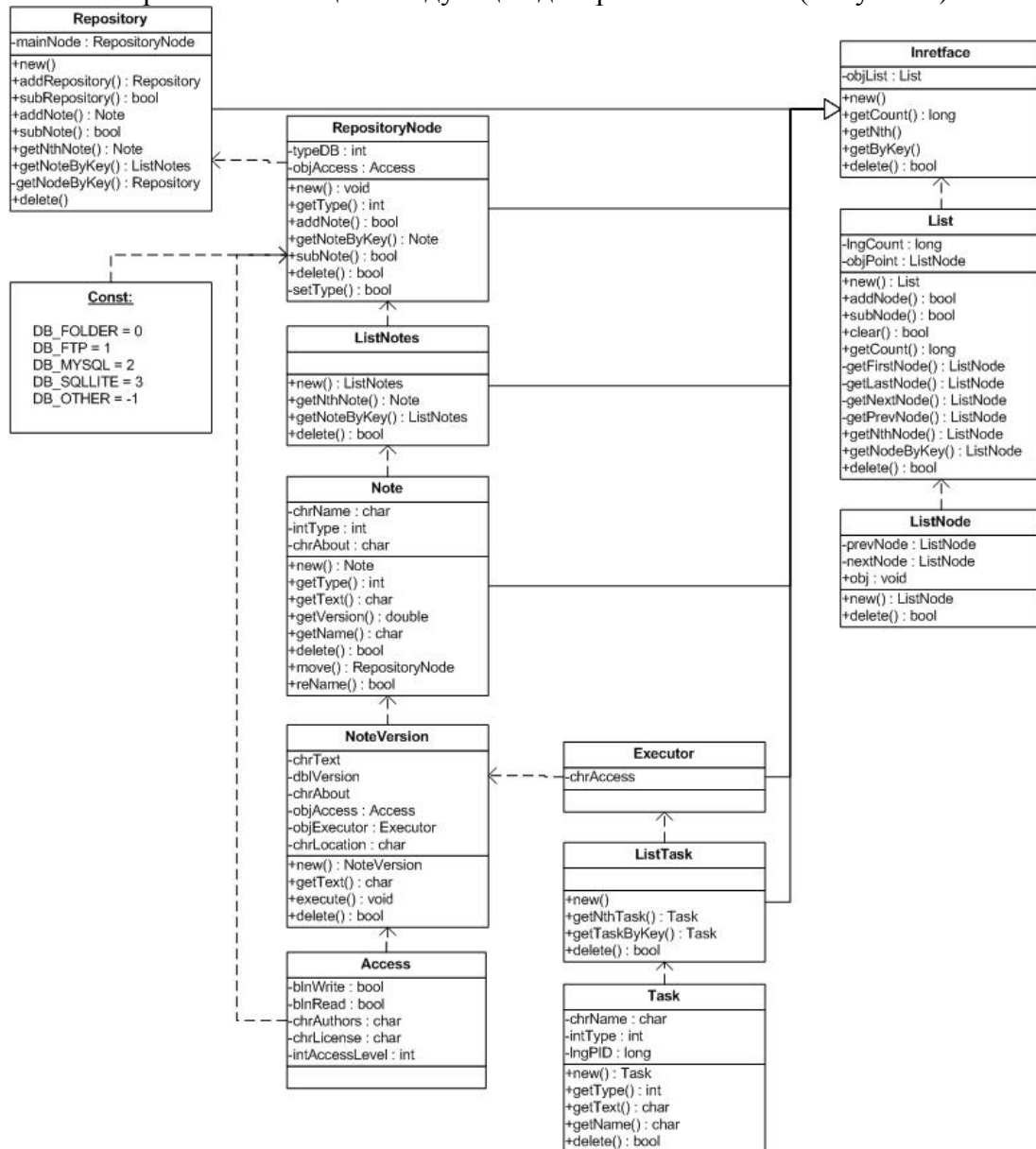


Рисунок 4. Диаграмма классов клиентской части системы.

Данная диаграмма состоит из следующих классов:

- Interface – класс для работы хранилищем данных, реализованным с помощью класса «List».
- List – хранилище, содержащее связанные между собой экземпляры класса «ListNode» и предоставляющее базовые методы их обработки.
- ListNode – узел списка, реализованного с помощью класса «List».
- Repository – базовый класс для работы со всем множеством репозитория, доступных пользователю, и предоставляющий возможность их поиска, добавления, удаления и т.д.
- RepositoryNode – репозиторий, предоставляющий доступ к хранящимся на нем данным в полном соответствии с политиками доступа.
- ListNotes – перечень хранящихся в репозитории данных (например, функций).
- Note – базовая единица хранящейся в репозитории информации.
- NoteVersion – версия экземпляра класса «Note».
- Access – класс, описывающий политику доступа к тому или иному элементу репозитория на чтение/запись данных и их выполнение на стороне сервера.
- Executor – класс, отвечающий за выполнение хранящихся в репозитории данных.
- ListTask – перечень выполняемых на стороне сервера процессов, запущенных удаленными или локальными пользователями репозитория.
- Task – процесс, запущенный локальным или удаленным пользователем репозитория.

Список констант указывает на возможность поддержки различных типов СУБД для хранения данных репозитория, а так же использования системы каталогов в том случае, если использование сторонних программных продуктов по какой-либо причине не представляется возможным.

Работа выполняется в рамках ФЦП НПП № 14.А18.21.0396 «Инструментальная поддержка архитектурно-независимой разработки параллельных программ на основе функционально-поточковой парадигмы параллельного программирования».

Список использованных источников:

1. Легалов А.И. Функциональный язык для создания архитектурно-независимых параллельных программ. / А.И. Легалов // Вычислительные технологии, № 1 (10) – 2005. - С. 71-89.