

Введение Компьютерные тренажерные комплексы занимают важное место в процессе подготовки персонала предприятий химической и нефтехимической отраслей [1]. Данные комплексы помогают поддерживать высокий уровень подготовки обслуживающего и управленческого персонала, что позволяет избегать ошибок с их стороны при проведении пусконаладочных операций, штатных остановах, нормальном ведении технологического процесса и что особенно важно при ликвидации нештатных аварийных ситуаций.

Необходимость наличия распределенных компьютерных тренажерных комплексов для проведения полномасштабных тренировок [2] под руководством инструктора [3] обусловлена как необходимостью использования современных средств обучения на предприятиях, так и предписаниями общих правил взрывобезопасности для взрывопожароопасных химических, нефтехимических и нефтеперерабатывающих производств (ПБ 09-540-03). Экономически целесообразно тренировки и обучение обслуживающего персонала проводить в составе рабочей группы или коллектива. Каждый обучаемый имеет свое рабочее место и индивидуальное задание. Для большей объективности взаимодействие обучаемых между собой исключается, а обмен информации происходит только с сервером тестирования (СТ). Преимущество распределенного комплекса заключается в возможности одновременной работы большого количества обучаемых как в специализированном компьютерном классе, так и удаленно при наличии доступа к серверу тестирования. При этом инструктор одновременно контролирует и управляет за ходом работы всех обучаемых. Имеется возможность удаленного доступа к серверу, что освобождает инструктора от территориальной привязанности к своему рабочему месту и позволяет вносить коррективы и осуществлять контроль за процессом обучения с любого компьютера с установленным программным обеспечением, имеющего доступ к СТ. В ходе разработки математического, алгоритмического и программного обеспечения был определен требуемый функционал распределенного тренажерного комплекса: - Среда для визуализации технологического процесса, содержащая базу данных компьютерных моделей и средств визуализации в количестве не менее 40 единиц; - Наличие инструментария для графической разработки технологических схем; - Наличие инструментария для разработки многопользовательской сетевой структуры; - Модуль разработки скриптового описания; - Модуль разработки процедур пуска, останова, локализации и ликвидации аварийных ситуаций; - Модуль разработки базы пользователей с возможностью разграничения прав доступа; - Наличие станции оператора; - Наличие станции инженера [4,5]. Структура тренажерного комплекса Для реализации принципа распределенности программного обеспечения в тренажерном комплексе реализована трехуровневая архитектура.

Трёхуровневая архитектура — архитектурная модель программного комплекса, предполагающая наличие в нём трёх компонентов: клиентского приложения

(обычно называемого «тонким клиентом» или терминалом), сервера приложений, к которому подключено клиентское приложение, и сервера базы данных, с которым работает сервер приложений. Клиент — это интерфейсный компонент, который представляет первый уровень, собственно приложение для конечного пользователя. Первый уровень не должен иметь прямых связей с базой данных, быть нагруженным основной бизнес-логикой и хранить состояние приложения. На первый уровень может быть вынесена и обычно выносятся простейшая бизнес-логика: интерфейс авторизации, алгоритмы шифрования, проверка вводимых значений на допустимость и соответствие формату, несложные операции (сортировка, группировка, подсчет значений) с данными, уже загруженными на терминал. Сервер приложений располагается на втором уровне, на котором сосредоточена большая часть бизнес-логики. Сервер базы данных обеспечивает хранение данных и выносятся на третий уровень. Обычно это стандартная реляционная или объектно-ориентированная СУБД. Если третий уровень представляет собой базу данных вместе с хранимыми процедурами, триггерами и схемой, описывающей приложение в терминах реляционной модели, то второй уровень строится как программный интерфейс, связывающий клиентские компоненты с прикладной логикой базы данных. Программный комплекс, структурная схема которого представлена на рис. 1 состоит из следующих структурных единиц: Программа ученик – программа для выбора теста и его прохождения. Программа учитель – программа для просмотра и контроля прохождения тестов учениками. Сервер тестирования – работает с учителем и учениками, осуществляет сохранение и выдачу данных для клиентских программ. Конструктор тестов – позволяет создавать сценарии для прохождения тестов. Модуль импорта – созданные тесты в конструкторе переносит в базу и регистрирует весь набор тестов. Рис. 1 - Схема работы программного комплекса

Рассмотрим механизм работы показанной схемы и содержание каждого элемента. Начальным элементом является конструктор тестов [6]. Для унификации с уже разработанными локальными тренажерными комплексами [7] данный элемент не претерпел изменений и включает в себя: - Среду для визуализации технологического процесса, содержащую базу данных компьютерных моделей и средств визуализации; - Наличие инструментария для графической разработки технологических схем; - Модуль разработки скриптового описания; - Модуль разработки процедур пуска, останова, локализации и ликвидации аварийных ситуаций; Это позволяет сохранять преемственность с уже разработанными упражнениями, моделями и графической оболочкой [8]. Конструктор тестов не связан программно или физически с остальными составляющими комплекса. Упражнения разрабатываются на любом ПК с установленным конструктором. После разработки необходимого набора тестов при помощи модуля импорта все тесты заносятся в базу данных, реализованную на базе реляционной системы

управления базами данных Microsoft Office Access. Данная СУБД имеет широкий спектр функций, включая связанные запросы, связь с внешними таблицами и базами данных. Благодаря встроенному языку VBA, в самой Access можно писать приложения, работающие с базами данных. Физически база данных хранится на компьютере, играющем роль сервера, программное обеспечение которого отвечает за взаимодействие клиентов (учеников и инструктора) друг с другом и с базой данных. Пользовательский интерфейс Для входа в систему в качестве инструктора используется специальное программное обеспечение, которое устанавливается на необходимом компьютере. При запуске необходимо пройти аутентификацию, т.е. ввести имя пользователя и пароль, которые задаются в базе данных. После прохождения процедуры аутентификации на экране появляется главное окно рис. 2. Рис. 2 - Окно инструктора системы Инструктор имеет возможность добавлять группы пользователей, например операторы/технологи/аппаратчики в меню группы. В меню пользователи отображается список учеников, имеющих возможность проходить обучение. В меню задания на тестирование отображается список всех имеющихся вариантов тестов. В поле сессии тестирования отображаются все пройденные тестирования зарегистрированных пользователей. Указывается название схемы, дата начала и окончания тестирования. При установке галочки в поле активные, видимыми остаются только сессии тестирования, проходящие в данный момент времени. В нижней части окна отображается перечень выполненных действий выбранного теста, идентичный отчету, конфигурируемому при окончании тестирования в окне прохождения тестирования [7]. Следующим логическим элементом системы является модуль ученика. Для прохождения тестирования необходимо запустить соответствующее приложение и пройти аутентификацию. При этом модуль ученика запрашивает у сервера тестирования (СТ) список учеников. После обработки запроса СТ отправляет сформированный список ученику. После выбора идентификатора ученика и ввода пароля ученик отправляет на СТ эти данные. При этом ученик формирует запрос на список разрешенных заданий. После проверки данных на правильность СТ в соответствии с таблицей прав формирует список задач и отправляет ученику. Ученик выбирает задачу из представленного списка (рис. 3), после чего на СТ отправляется данная информация. СТ совершает проверку режима тестирования (обучение/экзамен), после чего собирает данные для проведения тестирования и отправляет ученику следующую информацию: дата проведения тестирования, объекты с параметрами, линии связи, визуальные объекты, схема теста. Рис. 3 - Окно выбора задания После нажатия кнопки Старт ученик отправляет на СТ эту команду. Поле статуса меняет свое значение на активный. В таблице результатов создаются нулевые параметры. При совершении каждого действия ученик отправляет на СТ текущее состояние параметров, а СТ изменяет таблицу результатов. Завершение тестирования возможно в двух случаях: при нажатии

кнопки Стоп учеником и при окончании всех действий. При этом ученик отправляет команду завершения тестирования на СТ, а в строке состояния статус меняется на завершённый. Все запросы клиента (ученик/учитель) к СТ осуществляются посредством http протокола командой GET. Архитектура сервера Сервер получает от Клиента запрос GET, ядро разделяет строку с адресом и переменные запроса. В соответствии с адресом запроса выбирается контроллер и ему передаются переменные запроса. Контроллер выбирает действие и выполняет его. Действие контроллера обращается к модели, которая в свою очередь делает запрос к БД. Работа с БД возможна как из модуля инструктора, так и непосредственно в Microsoft Access. Графическое представление архитектуры сервера (рис. 4) содержит следующие логические элементы: ядро, контроллер, действие, модель, сервер, база данных. Рис. 4 - Архитектура сервера Практические результаты На основе разработанного программного обеспечения [6], математических моделей [8] были реализованы распределенные компьютерные тренажерные комплексы для обучения операторов действующих химических и нефтехимических производств [9,10].