

Добрый день!

Меня зовут Богомоллов Юрий, и я представляю проект ГПО АСУ-1101: "Облачная информационная система обучения студентов". Следует уточнить, что это уже новое название проекта, которое и относится к системе, которую я сейчас буду описывать.

Как вы, вероятно, помните, в прошлом семестре мы с бывшим коллегой разрабатывали концепцию обучающей системы, которая бы позволяла обучать студентов спортивному программированию.

Но прошёл семестр, появились новые интересные идеи, и я значительно переработал и улучшил концепцию разрабатываемой системы. Соответственно, поменялись цели и задачи проекта. Прежде под "системой" подразумевалась система, которая содержала бы набор статей и тестов по спортивному программированию.

Но зачем ограничиваться только спортивным программированием? Зачем обделять физику, математику, психологию, географию, социологию, химию, биологию, литературу? Верно, незачем. Поэтому первая цель - никаких ограничений по дисциплинам!

Помимо этого претерпела изменения и концепция учебного материала. Теперь это не просто какие-то текстовые статьи и тестики, а именно учебный материал, который можно интерпретировать с помощью интерпретаторов. Таким образом, можно создать такие интересные штуки, как интерактивные и

связанные между собой статья, картинки и исходный код. Поэтому вторая цель звучит так: разнообразие и интерактивность учебного материала.

Ещё изменения коснулись и тестов и задач. Они теперь, собственно, и есть учебный материал, который интерпретируется так, как нужно. Но это не просто учебный материал, а такой, который подразумевает ответ студента. Поэтому нужны обработчики решений студентов. Соответственно, третья цель такова: система может проверять решения студентов.

Исходя из всего вышесказанного, на семестр была поставлена следующая задача: разработать архитектуру системы, содержащую интерпретаторы учебного материала и обработчики решений студентов.

Кроме того, на следующие семестры уже можно определить следующие задачи:

- 1) начать разработку программной реализации системы;
- 2) разработать базовый набор интерпретаторов и обработчиков;
- 3) наполнять систему учебными материалами;
- 4) начать проектирование автоматизированного бота, способного самостоятельно выдавать задания студентам на основе их знаний.

Теперь давайте взглянем на упрощённую схему системы. На ней я продемонстрирую принцип взаимодействия системы и пользователей.

Для начала, преподаватель создаёт учебный материал. Из него он формирует задание и отправляет студенту или группе студентов на изучение и прорешку. У студентов на странице задание отображается с помощью

интерпретаторов. Затем студенты его прорешивают и отправляют обратно в систему. Обработчик проверяет решение, даёт по нему промежуточный вердикт и отправляет его вместе с решением преподавателю. Последний уже на основе вердикта системы и самого решения выставляет студентам окончательные оценки. При этом разнообразие и универсальность системы обеспечиваются за счёт различных обработчиков, интерпретаторов и конструкторов, создаваемых разработчиком.

Теперь перейдём к примерам того, какие проблемы можно будет решить с помощью этой системы.

Проблема первая. Точка зрения школьного учителя русской литературы.

Одной из форм проверки знаний здесь являются эссе и сочинения. Но какова вероятность того, что это сочинение не будет списано с какого-нибудь сайта с готовыми домашними заданиями, или не будет собрано из кусков краткого содержания, коих много в интернете? Ответ: вероятность не особо большая, хоть и не мизерная.

Решение же состоит в следующем. Учитель формулирует тему сочинения и отправляет его через интерфейс системы всему классу. Школьники получают задание, пишут сочинение и отправляют в систему. Система проверяет сочинение на плагиат, беря в качестве эталонов те же сочинения с сайтов готовых домашних заданий. Конечно, нужно учитывать, что не все быстро набирают текст на компьютере, поэтому будет предусмотрена возможность распознавать рукописный текст, написанный, например, в школьной тетради, и уже затем проверять его.

Проблема вторая. Взгляд со стороны студента, изучающего теорию графов.

Исследуется, например, алгоритм Дейкстры. Студент открывает статью на Википедии и что он видит? Он видит несколько экранов однообразных текста и картинок, пошагово демонстрирующих ход алгоритма, в то время как сам алгоритм находится где-то та-а-ам, в самом низу страницы. И где-то та-ам же представлен исходный код на одном лишь псевдокоде. Конечно, почти запросто можно взять и по-другому переписать эту статью, сделать её более наглядной и добавить примеров. Но, на мой взгляд, всё равно не хватает интерактивности и, может, некоторой компактности.

Я бы предложил, например, следующее решение: создать интерпретатор, который бы пошагово исполнял исходный код, показывал значения тех или иных переменных и отображал все изменения на картинке рядом.

Проблема третья. Разберём работу преподавателя по математике.

Во время учебного процесса необходимо периодически проверять знания студентов. Один из вариантов проверки - выдача десятка задач по, допустим, интегралам. Но если студентов много, то процесс проверки их решений становится нетривиальной и очень ресурсоёмкой и времязатратной задачей. Иногда это может занимать несколько дней, если не недель.

Преподаватель составляет задачи в редакторе формул или использует генератор задач, формирует задание и выдаёт его студентам. Студент после получения задания используют тот же редактор формул или распознаватель рукописного ввода, затем отправляет его в систему. Система анализирует решение студента и выдаёт вердикт, который преподаватель может использовать для выставления оценок студентам.

Теперь я бы хотел немного рассказать о технической части проекта.

Для начала хотелось бы уточнить, что разработку программной реализации я начну уже на летней практике. Так как интерфейс этой системы будет предоставляться в виде сайта, будет две части: серверная и клиентская. Клиентская часть будет писаться на связке HTML + JavaScript + CSS. Серверная же часть, вероятно, будет писаться на Python 3. Какие-то требовательные к скорости части будут писаться на C++.

Основной средой разработки я предполагаю выбрать Sublime Text 3. Для реализации на C++ будет использоваться MS Visual Studio.

И в заключение хотелось бы рассказать о планах на будущее. Во-первых, сейчас находится в разработке идея систем подсчёта рейтинга, которую лягут в основу автоматизированных ботов, способных выдавать задания студентам на основе знаний последних. Во-вторых, стоит уточнить, что процесс разработки программной реализации начнётся не в следующем семестре, а уже на летней практике.

На этом моё выступление заканчивается. Благодарю за внимание и  
ожидаю ваши вопросы.