



УДК 377.131.11; 377.131.14; 377.169.3

Особенности программной реализации цифровой адаптивной платформы для обучения программированию с функцией формирования навыков командной работы

Попков С.И.*

Московский государственный психолого-педагогический
университет, г. Москва, Российская Федерация
ORCID: <https://orcid.org/0000-0003-2566-1262>
e-mail: rslw25@gmail.com

Проведено исследование имеющихся подходов к построению образовательных сред для обучения программированию. Проанализированы их основные недостатки. Предложена концепция цифровой адаптивной платформы для обучения программированию, учитывающая выявленные недостатки. Описаны особенности программной реализации в виде общей архитектуры, принципов работы и построения взаимосвязей компонентов платформы. Продемонстрированы особенности программной архитектуры и реализации цифровой адаптивной платформы для обучения программированию с функцией формирования навыков командной работы.

Ключевые слова: адаптивность, цифровая платформа, обучение программированию, навыки командной работы.

Для цитаты:

Попков С.И. Особенности программной реализации цифровой адаптивной платформы для обучения программированию с функцией формирования навыков командной работы // Моделирование и анализ данных. 2020. Том 10. № 3. С. 85–100. DOI: <https://doi.org/10.17759/mda.2020100307>

***Попков Сергей Игоревич**, кандидат физико-математических наук, доцент факультета информационных технологий, заведующий лабораторией, Московский государственный психолого-педагогический университет, г. Москва, Российская Федерация, ORCID: <https://orcid.org/0000-0003-2566-1262>, e-mail: rslw25@gmail.com



ВВЕДЕНИЕ

Вопрос разработки и внедрения специализированных программных средств для обучения программированию актуален с конца XX века, с момента внедрения программирования в частности и информатики в целом как дисциплины, преподаваемой в учебных заведениях разного уровня – в школах, колледжах, высших и других учебных заведениях. Такие средства ставят своей задачей реализовать надежную с точки зрения информационной безопасности среду, при использовании которой обучающийся использует язык программирования (как правило, с ограниченным подмножеством доступных операторов и функций из некоторого реального языка) для воздействия на вычислительную среду и достижения заданного результата, тем самым закрепляя необходимые знания о процессе программирования и практике его применения на задачах, приближенных к реальным.

Существуют два наиболее часто используемых подхода к разработке подобных обучающих средств. Один из них основан на создании ряда предъявляемых заданий, оформленных в виде уровней с линейно растущей сложностью. Как правило, действия на каждом из таких уровней заранее ограничены некоторым набором возможных операций, чтобы упростить задачу обеспечения информационной безопасности и надежности обучающей среды. Примером такой среды является игра «Кодвардс» [1].

Второй подход подразумевает генерируемый свободный мир (также именуемый «песочницей», англ. “sandbox”), в котором обучающийся постепенно расширяет круг своих возможностей взаимодействия с миром за счет решения все более сложных задач. На финальном этапе обучающемуся доступен максимально широкий спектр возможностей для решения самой сложной задачи либо конечная цель задается через комплексную совокупность решения ряда более простых задач. Таким образом, мир на программном уровне обеспечивает полный набор возможных операций над ним с самого начала взаимодействия и взаимосвязанные задачи разного уровня сложности, но они закрыты для участника образовательного процесса до тех пор, пока он не разблокирует доступ к ним через решение задач соответствующего уровня сложности. Примером такой платформы является игра «CoLoBot» [2].

И в том, и в другом случае построение обучающей среды с обеспечением всех мер информационной безопасности, разработка транслятора для предоставляемого пользователю подмножества языка программирования, создание и оцифровка сценариев и графических элементов представляет собой набор достаточно трудоемких задач с высоким уровнем сложности их разработки как по отдельности, так и в качестве взаимосвязанных компонентов в составе обучающего комплекса, поэтому зачастую в таких средствах обучения нет элемента командного взаимодействия, который вносит свой вес в каждую составляющую критерия сложности разработки. Тем не менее, последние прорывы в области глубинного обучения [3] позволяют актуализировать вопрос разработки такой среды, представляющей собой цифровую платформу для обучения программированию с функцией формирования навыков командной работы.



Кроме того, зачастую задания, представленные в подобных платформах, строятся по принципу линейного увеличения сложности без учета реальных способностей обучающегося: предполагается что, если пользователь платформы прошел более легкое задание, он готов к следующему поступательно предъявляемому сложному, что не всегда верно. Возникает необходимость в применении адаптивного подхода к предъявлению заданий. Механизм, реализующий такой подход, был продемонстрирован в тренажере для операторов сложных систем [4, 5], однако он был ориентирован на предъявление заданий различного уровня сложности без учета характера ошибок, допущенных оператором. В данной работе необходимо преодолеть это ограничение алгоритма, поскольку игнорирование характера ошибки не позволяет обучающемуся проработать соответствующие разделы дисциплины, связанные с ними навыки и компетенции, что ведет к некачественному уровню понимания фундаментальных понятий дисциплины и образовательного процесса в целом.

Цель данной работы – исследовать имеющиеся подходы к построению образовательных сред для обучения программированию, выявление их недостатков и предложение собственного решения, позволяющего нивелировать или сгладить выявленные недостатки.

Актуальность материала, представленного в данной статье, обусловлена не только востребованностью решения для поставленной задачи, но и предлагаемыми способами практической реализации дополнительных возможностей, востребованных в образовательной сфере, таких как адаптивность предъявляемых заданий и формирование навыков командной работы у обучающихся.

ОБЗОР ПОПУЛЯРНЫХ ПОДХОДОВ К РЕАЛИЗАЦИИ СРЕДЫ ДЛЯ ОБУЧЕНИЯ ПРОГРАММИРОВАНИЮ

Применение цифровых обучающих технологий – один из актуальных подходов к ведению образовательного процесса, который принимает новые формы в связи с развитием программно-аппаратных средств и новейших стандартов веб-технологий. Появление современных систем управления курсами, как коммерческих, так и свободных (Moodle [6], edX [7], Chamilo [8] и др.), а также средств проведения тестирования (FlexiQuiz [9], ClassMarker [10], комплекс программных продуктов компании Pesofts [11]) позволяет реализовать гибкое окружение для ведения преподавательской деятельности и оценки уровня знаний студентов в различных сферах, таких, как математика или гуманитарные науки.

Однако перечисленные средства не учитывают специфику обучения программированию, которое представляет собой специфический процесс и требует определенных технологий, обеспечивающих оценку качества программного кода. Особенность такого рода программного обеспечения заключается в представлении взаимосвязанных заданий в игровой форме, призванной пробудить интерес к обучению, и форме ответа, представляющей собой свободный программный код, который выполняется в безопасной для вычислительной системы среде, автоматически проверяется на кор-



ректность и позволяет достичь поставленной в задаче цели множеством способов без необходимости выбора конкретного ответа или группы ответов (в отличие от классического подхода к тестированию).

Далее будут кратко рассмотрены наиболее известные и востребованные программные продукты, специализирующиеся на обучении программированию с учетом особенностей процесса обучения.

SberCraft

SberCraft – платформа, выпущенная группой компаний «Сбер» при поддержке компании «Gecko» [12]. Представляет собой игру в жанре RPG с линейным сюжетом, в которой на основе ряда параметров (языка программирования, уровня коммерческого опыта и т.д.). Цель – формирование компетенций, необходимых группе компаний «Сбер» для приема на работу, а также предварительная оценка квалификации кандидатов перед собеседованием. Может использоваться в качестве тренажера для программистов.

Среди сильных сторон можно отметить красивую реализацию, учитывающую целевую аудиторию, а также нетривиальные задачи с ограничением по времени их выполнения, богатая палитра языков (на момент написания статьи поддерживаются 12 языков программирования).

Индивидуальный недостаток: платформа заточена под рекрутинг и служит интересам конкретных компаний.

Недостатки, общие для всех программных продуктов, специализирующихся на обучении программированию, будут приведены ниже в разделе 2.

Кодвардс

Кодвардс – онлайн-платформа для обучения детей основам программирования в игровой форме. Разработана компанией «Redmadrobot» [13] с привлечением специалистов – детских педагогов, психологов, специалистов ИТ-индустрии. Представляет собой игру в двумерном пространстве, в которой агент (робот) решает задачи, выполняя команды упрощенного вымышленного языка программирования. Участник линейно переходит с уровня на уровень, поступательно увеличивая сложность и осваивая новые концепции программирования.

Среди сильных сторон можно отметить высокое качество графики, интуитивный интерфейс, подсказки, широкий охват материала, в том числе, основ объектно-ориентированной парадигмы программирования. Представлены задачи на поиск ошибок в коде.

Индивидуальный недостаток: платформа и ее содержимое ориентированы на учеников младшей школы.

CodinGame

Онлайн-платформа, разработанная одноименной компанией, «CodinGame» предлагает набор различных соревнований, нацеленных на совершенствование навыков программирования для всех уровней, включая опытных программистов [14]. Можно охарактеризовать как сборник стратегий в реальном времени с качественной трехмерной графикой, где для решения поставленных задач необходимо писать код для агента-исполнителя на реальном языке программирования.



Сильных сторон много, в их числе широкая палитра реальных языков программирования, множество предлагаемых тем, от простейших (например, циклы и условия) до сложных, ориентированных на опытных программистов (например, задачи на оптимизацию и многоагентные системы). Существует собственная система рекрутинга, позволяющая компаниям отбирать самых перспективных программистов [15]. Содержимое платформы постоянно пополняется новыми заданиями.

Индивидуальный недостаток: представляет собой набор разрозненных независимых заданий, не позволяющих учесть особенности учебной траектории лица, проходящего обучение.

CoLoBot

CoLoBot – (Colonize With Bots) – игра 2001 года, проходящая в данный момент процесс ремастеринга [16] (переиздания с учетом требований к современным программным продуктам). Это учебная стратегия в реальном времени для изучения программирования, включающая в себя миссии по колонизации различных планет с помощью роботов, управляемых программами на вымышленном языке программирования, близком по синтаксису к C++.

Сильные стороны проекта: свободный трехмерный мир, возможность создавать собственные карты (сценарии игры), интересные миссии (с единой сюжетной линией).

Индивидуальные недостатки: проект серьезно устарел, а предварительный релиз современного переиздания выполнен в феврале 2019 года (то есть на момент написания статьи прошло больше 1,5 лет с момента последнего предварительного издания, разработка идет достаточно медленно, последняя версия – 0.1.12). Продукт ориентирован на подростков, интересующихся программированием. Ограничения на объем входного кода не позволяют использовать серьезные возможности программирования, например, применение технологий искусственного интеллекта. Игра предусматривает режим, в котором может быть пройдена и без применения навыков программирования.

КЛЮЧЕВЫЕ НЕДОСТАТКИ РАССМОТРЕННЫХ ПОДХОДОВ

У рассмотренных проектов есть ряд существенных недостатков, которые важно устранить для более эффективного применения и развития образовательных платформ для обучения программированию.

Во-первых, ориентир на базу готовых заданий. Когда задания не генерируются и не модифицируются, становятся менее значимыми преимущества среды для написания произвольного кода, поскольку появляется возможность использовать чужой код у другого участника (например, студента старших курсов), то есть такой же «готовый ответ», как и в случае тестирования.

Во-вторых, линейность развития персонажа, с которым ассоциирует себя обучающийся. Сначала предъявляются наиболее простые задачи, затем – сложные. При этом необходимо пройти предъявляемую задачу текущего уровня сложности, чтобы выйти на следующий. Это, с одной стороны, приводит к снижению уровня вовлечен-



ности и интереса у более опытных участников на начальном этапе, а также создает риск «ловушки», когда участник не может пройти более сложное задание и продвигнуться вперед, что может привести к «выгоранию». Было бы полезно в таких случаях предусмотреть адаптивные упражнения, которые позволили бы преодолеть «разрыв» между уровнями: проанализировать неверные решения участника и отработать конкретные ситуации, вызывающие сложность (далее этот принцип для краткости будет называться «микроадаптивность»).

В-третьих, преимущественно индивидуальный подход в оценке уровня подготовки, отсутствие задач коллективного типа с ориентацией на развитие навыков работы в команде, что особенно важно для любого современного специалиста в IT-индустрии.

В-четвертых, сетевые проекты требуют больших затрат на безопасность, чтобы избежать «утечки» банка данных или нежелательного изменения хранилища данных с результатами работы участников. Для работы в аудитории есть возможность существенно упростить архитектуру без снижения уровня безопасности, что, в свою очередь, позволяет уделить время и другие ресурсы на разработку более полезных элементов платформы, например, функций формирования навыков командной работы.

В-пятых, в дистанционных платформах обучения отсутствует очное взаимодействие с преподавателем. Удобное переносимое архитектурное решение, допускающее локальное развертывание платформы, позволило бы не только отслеживать факты мошенничества и злоупотребления платформой со стороны студентов, но и, по усмотрению преподавателя, допускать его участие и советы по решению конкретного участка задачи. При этом, при необходимости, программная архитектура платформы допускает возможность перевода состояния среды в дистанционный режим.

В-шестых, статичные задания не только увеличивают вероятность списывания, но и не позволяют построить систему, которая адаптивно подбирает бы траекторию задач и с учетом уровня конкретного обучающегося индивида, и с учетом уровня группы в целом. Тогда как правильно обученная модель машинного обучения может более тонко подогнать динамически изменяемые параметры на основе статистических данных об успехах конкретного студента, сделав переход между уровнями навыков менее дискретным.

В-седьмых, платформы не всегда учитывают особенности студентов с ограниченными возможностями для равноправной оценки уровней всех групп студентов. К примеру, для защиты от копирования условие задачи может быть представлено в форме монолитного изображения, недоступного для распознавания программными средствами для лиц с нарушениями зрения. В подобной ситуации, при индивидуальном подходе, можно учесть особенности и представить такому лицу подходящую по форме задачу того же уровня. Это может быть особенно актуально для инклюзивных групп в режиме командной работы. Для этого должен быть реализован редактор, позволяющий вносить изменения в генерируемые блоки заданий и их предпросмотр.

Для того, чтобы устранить выявленные недостатки, необходимо создать более комплексную платформу с особыми характеристиками.



ПОСТАНОВКА ЗАДАЧИ И ФОРМУЛИРОВАНИЕ УСЛОВИЙ, КОТОРЫМ ДОЛЖНА СООТВЕТСТВОВАТЬ ЦИФРОВАЯ АДАПТИВНАЯ ПЛАТФОРМА ДЛЯ ОБУЧЕНИЯ ПРОГРАММИРОВАНИЮ С ФУНКЦИЕЙ ФОРМИРОВАНИЯ НАВЫКОВ КОМАНДНОЙ РАБОТЫ

С учетом исследованных решений можно сформулировать цель разрабатываемого проекта: создать адаптивную учебную платформу с функцией формирования навыков командной работы. Платформа в формате тематической электронной викторины позволяет оценить и повысить уровень навыков студентов (участников) и группы в целом через решение предъявляемых задач, которые имеют динамически изменяющиеся компоненты и используют механизмы адаптивного обучения. Платформа должна удовлетворять ряду условий и принципов, устраняющих обнаруженные ранее недостатки у прочих платформ подобного рода.

- 1) Модульность – платформа представляет собой систему с модульной архитектурой и иерархией вложенных представлений учебного материала, каждое из которых реализует свой уровень абстракции (курс, лекция, тема, уровень и конкретная задача). Такая гибкая система допускает современный компонентно-ориентированный подход к программной реализации и позволяет создавать нелинейные связи и переходы между отдельными заданиями при построении образовательной траектории группы. Кроме того, появляется возможность создания форм представления заданий для лиц с ограниченными возможностями.
- 2) Микроадаптивность – каждая конкретная задача обладает микроиерархией простейших элементов, реализующих механику для оттачивания конкретного навыка, задействованного в задаче. Эти элементы могут быть также распределены по уровням. Допущенные студентом ошибки при решении участка задачи позволяют определить сегмент знаний, в котором наблюдаются пробелы, а также более тонко корректировать уровень сложности, чтобы участник в ненавязчивой форме мог последовательно отрабатывать и исправлять те ошибки, которые свойственны именно ему.
- 3) Определение входного порога – платформа способна осуществлять адаптивное обучение, подстраиваясь под уровень студента. На входном этапе при регистрации в системе студент проходит адаптивное тестирование, позволяющее в первом приближении оценить уровень навыков. Далее система сама анализирует и корректирует, на основе статистики прохождения этапов викторины, текущий уровень студента.
- 4) Тематическое оформление – задания как таковые представляют собой абстрактные компоненты, которые необходимо заставить корректно работать, чтобы получить требуемую информацию или обеспечить победу в викторине. Для того, чтобы задания можно было использовать в конкретном окружении (тематическом виртуальном пространстве заданий), используются декораторы, которые оформляют механику заданий в форму, привлекательную для участника викторины, используя специальный внутренний декларативный язык программирования,



позволяющий оперативно и эффективно описывать динамически изменяющиеся структуры в элементах предъявляемых задач.

- 5) Навыки командной работы – помимо отдельных заданий, позволяющих установить индивидуальный уровень каждого участника, платформа может оценить способности группы в целом и использовать специальные групповые окружения, где из отдельных компонентов собирается уникальная траектория заданий таким образом, чтобы каждое задание соответствовало уровню выбравшего его участника, при этом результаты отдельных заданий пересекаются, побуждая участников к командному взаимодействию для достижения конечной цели, поставленной групповым окружением. Это позволит не только более точно оценить уровень группы и подбирать более подходящие задания на следующих этапах викторины, но и способствовать формированию навыков командной работы и сплоченности группы в целом (что особенно важно для таких курсов, как, например, программная инженерия).
- 6) Единый дружественный интерфейс для комплекса программ – внешний интерфейс, объединяющий компоненты программного комплекса, обслуживающего адаптивную учебную платформу, должен предоставлять групповую и индивидуальную статистику результатов прохождения этапов викторины в удобной для восприятия преподавателем форме. Впоследствии можно обеспечить и визуальный конструктор или интерактивную среду разработки для упрощения процедуры создания компонентов курса, в частности, учесть специфику разработки и поддержки заданий для лиц с ограниченными возможностями.
- 7) Гибкость реализации – программная реализация платформы должна основываться на принципах переносимости и кроссплатформенности. Необходимо предусмотреть возможность быстрого локального развертывания системы и удобства вызова основных функций, обеспечивающих работу платформы и ее использование для разных программно-аппаратных сред.

КОМПОНЕНТЫ И ОБЩАЯ АРХИТЕКТУРА ЦИФРОВОЙ АДАПТИВНОЙ ПЛАТФОРМЫ

Платформа представляет собой большой программный комплекс, состоящий из не менее трех взаимосвязанных серверов (которые могут быть реализованы с применением технологий виртуализации), не менее двух хранилищ данных и локальной сети, используемой в процессе обучения для осуществления распределенного доступа к рабочей среде и упражнений, формирующих навыки командной работы. Каждый сервер обеспечивает работу набора служб, соответствующих сфере его применения, а также доступ для пользователей, соответствующих роду выполняемых средствами платформы работ (обучение, описание компонентов предъявляемых заданий, сбор и анализ статистики и т.д.). Представление схемы основных компонентов занимает слишком внушительный размер, чтобы ее можно было отобразить в одном рисунке. Общее представление схемы показано на рис. 1, далее будет представлена подробная детализация каждого обобщенного элемента.

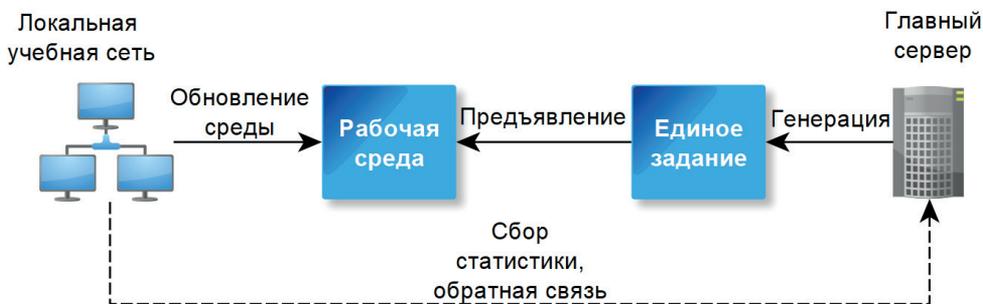


Рис. 1. Общее графическое представление схемы компонентов цифровой адаптивной платформы для обучения программированию с функцией формирования навыков командной работы

Так как фактическая реализация цифровой адаптивной платформы для обучения программированию с функцией формирования навыков командной работы осуществляется в рамках ВУЗа для студентов-бакалавров, далее будет использован термин «студент» для обозначения обучающегося пользователя; однако, стоит отметить, что программный комплекс может осуществлять свои функции и для других образовательных программ (например, в школах).

Набор компонентов, отвечающий за функционирование локальной учебной сети, хранит данные о студентах и группах, которые являются пользователями платформы. Эти данные используются для сопоставления уровня конкретного пользователя и группы в целом, проверки прав доступа на авторизацию и доступ к рабочей среде. Также этот набор компонентов осуществляет функции обновления рабочей среды за счет действий пользователя (например, загрузки валидного сегмента кода, требуемого для прохождения задания) и работу sandbox-сервера, обеспечивающего функции взаимодействия с рабочей средой на стороне клиента с учетом требований информационной безопасности, а также отправку данных на главный сервер о ходе обучения и взаимодействия с рабочей средой – в частности, статистики, необходимой для коррекции уровня подготовки обучающегося и обучения модели машинного обучения, отвечающей за формирование задания и окружения рабочей среды.

Рабочая среда – это виртуальное пространство, обеспечиваемое клиент-серверным взаимодействием между участниками образовательного процесса и главным сервером. Поэтому набор компонентов, обеспечивающий функционирование рабочей среды, носит распределенный характер. Со стороны главного сервера, формирующего рабочую среду, требуется предоставление основы задания, соответствующей текущему сценарию (набору параметров, обеспечивающих взаимосвязь компонентов заданий, и декоративных элементов, поддерживающих вовлеченность и заинтересованность студентов в прохождении заданий). Также на стороне сервера компоненты задания (или динамические составляющие) подбираются с учетом истории взаимодействия участников с рабочей средой и их уровня подготовки. На основе этих данных формируется единое задание и соответствующий ему набор состояний рабочей среды. Со стороны



обучающихся и промежуточных серверов, обслуживающих их взаимодействие с рабочей средой, требуется предоставить функции, обеспечивающие безопасный ввод данных, влияющих на состояние рабочей среды и решение поставленной задачи.

Набор компонентов, обеспечивающих работу главного сервера, представляет собой наиболее высоконагруженную часть всего программного комплекса, так как реализует функции сбора и анализа статистики, процедуры машинного обучения, ответственные за актуализацию учебных сценариев и распределение заданий для студентов и групп по уровням, операции обслуживания и наполнения банка заданий, состоящего из томов баз данных, хранящих сценарии и фрагменты предъявляемых в рабочей среде заданий, а также данные авторизации для лиц, ответственных за ручное наполнение и редактирование банка заданий – например, уполномоченных преподавателей и системных администраторов. Эта же группа лиц имеет доступ к статистике решений по группам и индивидуальной статистике, отражающей прогресс конкретного учащегося лица в изучении предмета средствами платформы. Также главный сервер выполняет свои задачи по формированию единого задания для рабочей среды, учитывающего уровень группы и индивидуальных учащихся.

Набор компонентов, отвечающий за функционирование локальной учебной сети, подробно описан на рис. 2.

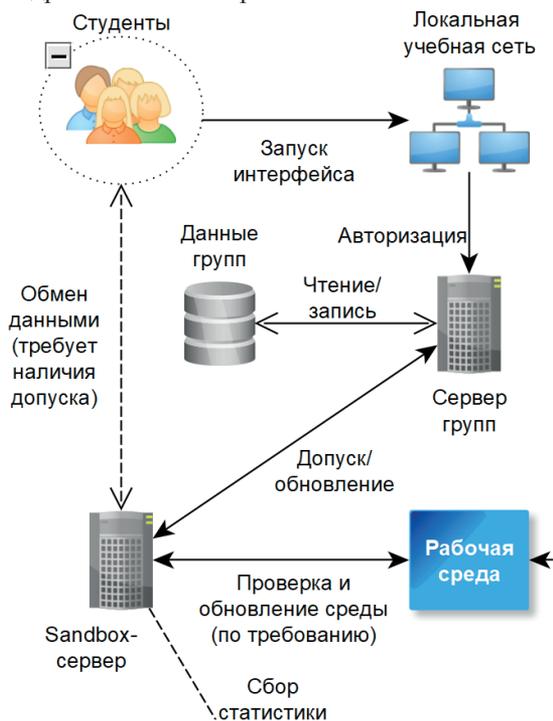


Рис. 2. Набор компонентов, соответствующий элементу «Локальная учебная сеть» на общем графическом представлении схемы компонентов

Студенты инициируют работу информационной системы, осуществляя попытку подключения к ее интерфейсу через локальную учебную сеть. Выполняется проверка корректности введенных пользователями данных для входа в систему, для этого применяется сервер групп, который сверяет вводимые данные с зашифрованным содержимым хранилища данных – базы данных о группах; в случае успешной авторизации сервер обеспечивает запуск сессии и допуск для взаимодействия с рабочей средой или обновление информации о текущем состоянии рабочей среды для авторизованных пользователей. Sandbox-сервер подготавливает рабочую среду, которая учитывает характеристики конкретного пользователя (включая его уровень подготовки). Для этого, с помощью полученного на основе данных ав-

торизации идентификатора, sandbox-сервер посылает запрос главному серверу в форме обратной связи и данных статистики о текущем состоянии студента в контексте рабочей среды. Информация, которая поступает в ответ на запрос от главного сервера, позволяет sandbox-серверу сконструировать индивидуальную рабочую среду для каждого студента с учетом его действий и действий прочих студентов в группе. Визуально рабочая среда отображается для студента на стороне клиента – например, браузера, где студент получает возможность осуществлять взаимодействие с рабочей средой в соответствии с предоставляемыми веб-интерфейсом возможностями. Подразумевается, что взаимодействие осуществляется через компьютер в аудитории, имеющий доступ к локальной учебной сети ВУЗа. Однако, принципы клиент-серверной архитектуры позволяют расширить применение программной платформы и для случая дистанционного обучения.

Набор компонентов, отвечающий за функционирование рабочей среды и предъявление единого задания, подробно описан на рис. 3.

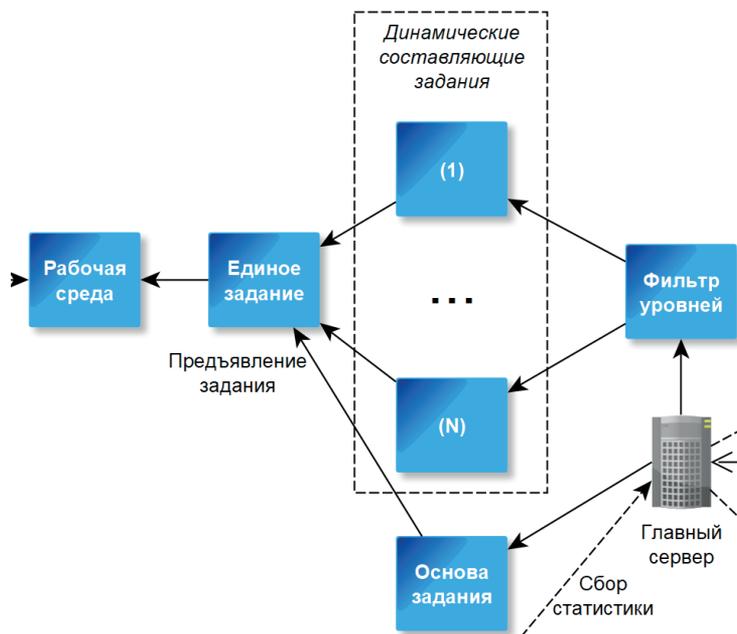


Рис. 3. Набор компонентов, соответствующий элементам «Рабочая среда» и «Единое задание» на общем графическом представлении схемы компонентов

Главный сервер, на основе собранной статистики и полученной обратной связи от sandbox-сервера, подбирает динамические составляющие – компоненты задания – таким образом, чтобы они соответствовали уровню подготовки студента и учитывали ошибки, допущенные в прошлом. На момент, когда тот или иной студент берется за выполнение индивидуального задания, он вкратце знаком только с его основой, описывающей сюжет и тему задания. Когда решение о закреплении индивидуально-



го задания принято, главный сервер, используя данные о студенте и фильтр уровней, модифицирует (подбирает) компоненты задания (детали сюжета и промежуточные задания) таким образом, чтобы они отвечали уровню подготовки студента и позволяли ему проработать допущенные ранее ошибки.

Основа задания задает сюжет для рабочей среды, в котором участвует вся группа. Сюжет и характер задания подбирается исходя из общего уровня группы, показанного на предыдущих занятиях (для первого занятия задается некоторый уровень по умолчанию). После того, как индивидуальные задания выполнены, для достижения общей цели студентам, выполнявшим индивидуальные задания, необходимо, в соответствии с сюжетом, объединить результаты своей индивидуальной работы и решить коллективное задание. Таким образом, формируются навыки командной работы, каждый студент может проявить свои сильные стороны в ходе решения индивидуальных заданий, а также устраняется проблема, при которой один или несколько способных студентов в группе выполняют всю работу за остальных: при наличии индивидуального задания и некоторой части общего для каждого участника не остается ресурсов, чтобы вникнуть и решить чужое индивидуальное задание (с учетом ограничения по времени в 2 академических часа).

После того, как все индивидуальные компоненты постановки задач подобраны и соединены в единое целое с основой задания, задающей сюжет, окружение и групповую задачу, из этих компонентов формулируется единое задание – скрипт, содержащий описание окружения, текстур и прочих декоративных элементов, а также сюжет и основные логические взаимосвязи между результатами индивидуальных заданий и общей для группы цели. На основе этого единого задания формируется рабочая среда и ответы на запросы sandbox-сервера.

Набор компонентов, отвечающий за функционирование главного сервера, подробно описан на рис. 4.

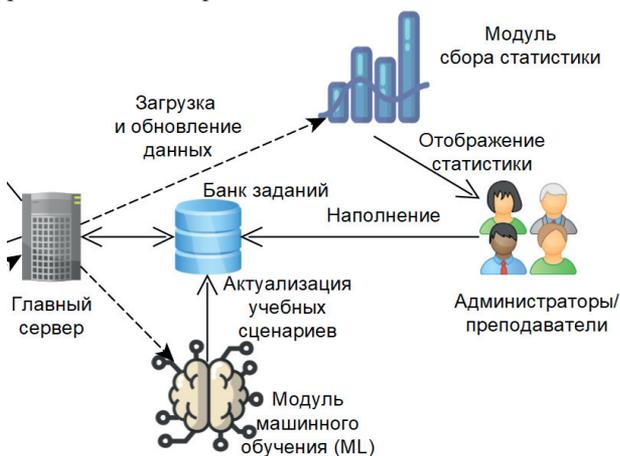


Рис. 4. Набор компонентов, соответствующий элементу «Главный сервер» на общем графическом представлении схемы компонентов

Главный сервер формирует основу задания и подготавливает индивидуальный фильтр уровней для каждого студента, авторизованного и имеющего доступ к учебной среде. На основе этих данных главный сервер выполняет свою часть работы по формированию рабочей среды. Статистические данные, передаваемые в качестве формы обратной связи от sandbox-сервера, главный сервер учитывает, фиксирует и передает в две независимые



службы, реализованные в форме модулей – сбора статистики и машинного обучения. Задача модуля сбора статистики – собрать данные и представить их в наглядной форме уполномоченному лицу, отвечающему за наполнение банка заданий (преподавателю, администратору и т.п.). Модуль машинного обучения использует технологии и модели искусственного интеллекта для коррекции и заполнения банка заданий с целью актуализации учебных сценариев на основе результатов работы студентов, реализуя, тем самым, компонент адаптивности. Лица, отвечающие за наполнение банка заданий, имеют возможность скорректировать результаты работы модуля машинного обучения и осуществлять самостоятельное наполнение банка заданий. Главный сервер также имеет прямой доступ к банку заданий для записи информации, идентифицирующей ход выполнения заданий группой студентов, а также извлечения данных о текущих заданиях и их компонентах для конкретной группы авторизовавшихся студентов.

ЗАКЛЮЧЕНИЕ

В статье были исследованы имеющиеся подходы к построению образовательных сред для обучения программированию. Проанализированы их основные недостатки. Предложена концепция цифровой адаптивной платформы для обучения программированию, учитывающая выявленные недостатки. Описаны особенности программной реализации в виде общей архитектуры, принципов работы и построения взаимосвязей компонентов платформы.

Описанная архитектура реализует все условия и принципы, позволяющие преодолеть выявленные у похожих продуктов недостатки:

- 1) модульность и микроадаптивность – реализованы за счет системы динамических составляющих задания;
- 2) определение входного порога – за счет модуля сбора статистики и функций sandbox-сервера;
- 3) тематическое оформление и единый дружественный интерфейс для комплекса программ – за счет модулей и функций главного сервера;
- 4) навыки командной работы – за счет создания виртуальной рабочей среды на базе клиент-серверной архитектуры и предъявления единого задания, учитывающего уровень подготовки группы и совокупность результатов выполнения индивидуальных заданий;
- 5) гибкость реализации – за счет компонентного подхода к построению цифровой адаптивной платформы для обучения программированию с функцией формирования навыков командной работы.

Таким образом, в статье были продемонстрированы особенности программной реализации цифровой адаптивной платформы для обучения программированию с функцией формирования навыков командной работы.

Литература

1. Кодвардс – URL: <https://codewards.ru> (дата обращения 10.12.2020).
2. CoLoBot for Windows (2001) – MobyGames – URL: <https://www.mobygames.com/game/colobot> (дата обращения 10.12.2020).



3. Гудфеллоу Я., Бенджио И., Курвилль А. Глубокое обучение / пер. с англ. А.А. Слинкина. – М.: ДМК Пресс, 2017. – 652 с.: ил.
4. Kuravsky L.S., Popkov S.I. and Artemenkov S.L. An applied multi-agent system within the framework of a player-centered probabilistic computer game. – International Journal of Modeling, Simulation, and Scientific Computing, Vol. 9, No. 1 (2018), 17 pp, DOI: 10.1142/S1793962317500635 [SCOPUS].
5. Kuravsky L.S., Popkov S.I. and Artemenkov S.L. Applied multi-agent system to study behavior of operators of complex technical systems. – In: Proc. First World Congress on Condition Monitoring 2017 (WCCM 2017) The International Society for Condition Monitoring (ISCM), British Institute of Non-Destructive Testing (BINDT). 2017 [SCOPUS].
6. Moodle – URL: <https://moodle.org> (дата обращения 10.12.2020).
7. edX – URL: <https://www.edx.org> (дата обращения 10.12.2020).
8. Chamilo LMS – URL: <https://chamilo.org/en/> (дата обращения 10.12.2020).
9. FlexiQuiz – URL: <https://www.flexiquiz.com> (дата обращения 10.12.2020).
10. ClassMaker – URL: <https://www.classmarker.com> (дата обращения 10.12.2020).
11. Pesofts – URL: <https://pesofts.com> (дата обращения 10.12.2020).
12. SberCraft – URL: <https://sber.geecko.ru> (дата обращения 10.12.2020).
13. Redmadrobot запускает платформу «Кодвардс» для обучения детей основам программирования – URL: <https://www.redmadrobot.ru/fyi/codewards> (дата обращения 10.12.2020).
14. CodinGame – URL: <https://www.codingame.com/start> (дата обращения 10.12.2020).
15. CodinGame for Work – URL: <https://www.codingame.com/work> (дата обращения 10.12.2020).
16. Official Colobot: Gold Edition website – URL: <https://colobot.info> (дата обращения 10.12.2020).



Features of the Software Implementation of the Digital Adaptive Platform for Learning Programming with the Teamwork Skills Forming Function

Sergei I. Popkov*

Moscow State University of Psychology and Education, Moscow, Russia

ORCID: <https://orcid.org/0000-0003-2566-1262>

e-mail: rslw25@gmail.com

A study of existing approaches to creating educational environments for learning programming is conducted. Their main shortcomings are analyzed. The concept of a digital adaptive platform for learning programming, taking into account the identified shortcomings, is proposed. The features of the software implementation are described in a form of links between platform components, operation concepts and the platform architecture overall. The features of the software architecture and implementation of the digital adaptive platform for learning programming with the teamwork skills forming function are demonstrated.

Keywords: adaptability, digital platform, learning computer programming, teamwork skills.

For citation:

Popkov S.I. Features of the Software Implementation of the Digital Adaptive Platform for Learning Programming with the Teamwork Skills Forming Function. *Modelirovanie i analiz dannyykh = Modelling and Data Analysis*, 2020. Vol. 10, no. 3, pp. 85–100. DOI: <https://doi.org/10.17759/mda.2020100307> (In Russ., abstr. in Engl.).

References

1. Codewards – URL: <https://codewards.ru> (req. date 10/12/2020)/
2. CoLoBot for Windows (2001) – MobyGames – URL: <https://www.mobygames.com/game/colobot> (req. date 10/12/2020)/
3. Ian Goodfellow, Yoshua Bengio, Aaron Courville. Deep Learning (in Russ.) – M.: DMK Press, 2017. – 652 pp.: il.
4. Kuravsky L.S., Popkov S.I. and Artemenkov S.L. An applied multi-agent system within the framework of a player-centered probabilistic computer game. – *International Journal of Modeling, Simulation, and Scientific Computing*, Vol. 9, No. 1 (2018), 17 pp, DOI: 10.1142/S1793962317500635 [SCOPUS].
5. Kuravsky L.S., Popkov S.I. and Artemenkov S.L. Applied multi-agent system to study behavior of operators of complex technical systems. – In: Proc. First World Congress on Condition Monitoring 2017 (WCCM 2017) The International Society for Condition Monitoring (ISCM), British Institute of Non-Destructive Testing (BINDT). 2017 [SCOPUS].

***Sergei I. Popkov**, PhD in Physical and Mathematical Sciences, Assistant Professor of the faculty of information technologies, head of the laboratory, Moscow State University of Psychology and Education, Moscow, Russia, ORCID: <https://orcid.org/0000-0003-2566-1262>, e-mail: rslw25@gmail.com



6. Moodle – URL: <https://moodle.org> (req. date 10/12/2020).
7. edX – URL: <https://www.edx.org> (req. date 10/12/2020).
8. Chamilo LMS – URL: <https://chamilo.org/en/> (req. date 10/12/2020).
9. FlexiQuiz – URL: <https://www.flexiquiz.com> (req. date 10/12/2020).
10. ClassMaker – URL: <https://www.classmarker.com> (req. date 10/12/2020).
11. Pesofts – URL: <https://pesofts.com> (req. date 10/12/2020).
12. SberCraft – URL: <https://sber.geecko.ru> (req. date 10/12/2020).
13. “Redmadrobot” company launches «Codewards» platform to teach children the basics of the programming – URL: <https://www.redmadrobot.ru/fyi/codewards> (req. date 10/12/2020).
14. CodinGame – URL: <https://www.codingame.com/start> (req. date 10/12/2020).
15. CodinGame for Work – URL: <https://www.codingame.com/work> (req. date 10/12/2020).
16. Official Colobot: Gold Edition website – URL: <https://colobot.info> (req. date 10/12/2020).