

УДК 004.774.6+37.016

ИНСТРУМЕНТЫ ПОДДЕРЖКИ РОЛЕВЫХ ЗАДАНИЙ ПО СТРАТЕГИИ STAD В ОБУЧАЮЩЕЙ СИСТЕМЕ

В. В. Матюнин¹, А. А. Марченко²

^{1,2} Высшая школа информационных технологий и интеллектуальных систем Казанского (Приволжского) федерального университета

¹vlad.matyunin96@gmail.com, ²anton.a.marchenko@gmail.com

Аннотация

Представлена одна из возможных реализаций модели совместного обучения по ролям, основанной на стратегии STAD (Student Teams-achievement Divisions) кооперативного обучения в LMS (Learning Management System, Система управления обучением). Подходы, описанные в данной образовательной методике, развивают у обучающихся навыки командной работы, необходимые в профессиональной деятельности, а их внедрение в систему обучения позволит автоматизировать и оптимизировать некоторые процессы и открыть новые возможности для реализации новых инструментов.

Ключевые слова: кооперативное обучение, STAD, LMS, обучающие системы

ВВЕДЕНИЕ

Большинство систем управления обучением (далее LMS – Learning Management System) акцентирует свое внимание на развитие прикладных навыков. Для специалистов программной инженерии примерами подобных навыков являются программирование на определенном языке и использование различных инструментов разработки. Также современные LMS содержат инструменты для формирования прикладных навыков: тестирование, онлайн-уроки с возможностью выполнять и проверять задания. С их помощью обучаемые могут закрепить пройденный материал, изучить пропущенные темы.

Безусловно, прикладные навыки необходимы для будущих специалистов. Однако от них также требуются гибкие навыки (soft skills): работа в команде, адаптация к новым требованиям и условиям. Одним из основных сценариев их

приобретения является коллективная работа. В работах [1, 2] отмечается, что самые распространенные LMS, такие, как Moodle, Blackboard, хотя и обладают достаточным функционалом (хранение и распределение материалов, тестирование, построение курсов и уроков), но не соответствуют современным стандартам обучения с совместным выполнением заданий, автоматизацией.

Несмотря на наличие некоторых коммерческих e-learning платформ с возможностью поддержки командной работы (Teamie, Projectfoundry, Mindscroll), пока не существует платформ, сфокусированных на ролевых заданиях. Их основной задачей является создание окружения, в котором студенты могут работать над проектом или задачей совместно. Также роли каждого из учащихся в таких системах не декларированы. Из этого можно сделать вывод, что на данный момент не существует LMS с поддержкой ролевых заданий.

МОДЕЛИ СОВМЕСТНОГО РОЛЕВОГО ОБУЧЕНИЯ

Среди большого множества моделей обучения с поддержкой совместной ролевой деятельностью наиболее популярными являются модели кооперативного, совместного и проектного обучения. Каждая из них по-своему представляет то, как учащиеся обучаются, решая определенные задачи в группах.

В работе [3] *проектное обучение* определяется как модель, которая организует образовательный процесс вокруг проектов, при этом под проектом подразумевается множество сложных задач, побуждающих учащихся планировать, исследовать и принимать решения. Проектное обучение также дает возможность студентам работать относительно независимо друг от друга в течение продолжительного времени. Работа над проектом мотивирует студентов, формирует у них чувство удовлетворенности, повышает ответственность.

В *кооперативном обучении* основной упор делается на взаимодействие студентов, которое не обязательно можно достичь с помощью групповых заданий. В 1994 году Роджер Джонсон и Давид Джонсон сформировали 5 основных терминов данной модели [4]:

1. Положительная взаимозависимость – принцип, при котором успех каждого из членов команды напрямую зависит от остальных, при этом учащийся не может преуспеть без вклада его коллег;
2. Индивидуальная ответственность – правило кооперативного обучения, когда каждый член команды несет ответственность за групповой результат.

Она проявляется тогда, когда происходит оценка работы учащегося и результаты возвращаются к нему и его команде;

3. Прямая поддержка необходима для реализации положительной взаимозависимости, то есть нужно мотивировать студентов для помощи своей команде, тем самым повышая общий успех команды;

4. Чтобы пункт, описанный выше, действительно работал, необходимо развивать в учащихся навыки общения;

5. Групповая работа проявляется, когда команда обсуждает текущий общий результат и результат каждого в команде индивидуально.

Совместное обучение – это модель, при которой двое или более студентов совместно изучают что-либо, при этом количество обучающихся может быть совершенно разным (группа 2–4 людей, большая группа или даже сообщество). При этом чаще всего основная учебная деятельность в данной модели сконцентрирована вокруг решения какой-либо задачи, а само обучение является лишь результатом данного подхода. При совместном обучении большую роль играют также сами задания: они должны быть построены на тесном взаимодействии внутри группы учащихся. Такой подход имеет как ряд плюсов, так и ряд минусов: из-за тесной связи внутри коллектива каждый из студентов демонстрирует и укрепляет свои способности работать сообща, разделять или нести ответственность, но при этом та же связь создает множество трудностей при оценке результатов (при совместном обучении оценивается результат всей команды в целом) и зависимость внутри команды (один из учащихся не может завершить свою задачу, так как она связана с задачами его коллег).

СТРАТЕГИЯ STAD

Стратегия STAD является одной из разновидностей кооперативного обучения и была сформирована американским психологом Робертом Славиним [5]. Основной ее идеей является работа в небольших командах из 4–5 участников с разными способностями, успеваемостью. Образовательный процесс при использовании данной стратегии можно описать следующим образом: в начале преподаватель формирует группы, после чего формирует задания для каждой из них. Далее происходит преподавание материала всем учащимся, которое закрепляется их командным заданием. После его выполнения происходит тести-

рование каждого из студентов лично, и на основе данных оценок формируется общий результат команды. Важно отметить, что в данном случае главным отличием такой методики от традиционного обучения является выполнение задания, а преподавание материала остается прежним (преподаватель рассказывает материал всем учащимся).

В работе [6] отмечается, что при STAD-стратегии студенты демонстрировали гораздо более высокие результаты на финальном тестировании, чем их коллеги, обучавшиеся методами традиционного обучения.

При формировании групп преподаватель может также обозначить роли внутри каждой из команд. Это не только позволит повысить общую эффективность команды, но и даст возможность студентам обучаться навыкам командной работы в фиксированной роли.

ПРЕДПОЛАГАЕМЫЙ ОБРАЗОВАТЕЛЬНЫЙ ПРОЦЕСС

Из-за простоты стратегии STAD ее легко реализовать в формате обучающей системы. Если первую часть обучения (преподавание теоретического материала) можно перенести на формат LMS, то для поддержки совместной работы над задачами необходимы специальные инструменты.

При формировании образовательного процесса внутри будущей платформы были сформированы два главных типа ее участников: преподаватель и студент. Задание для групп можно представить в виде проекта, который строго делится на временные периоды. В течение каждого из таких периодов преподаватель вносит общее большое задание, которое студенты преобразуют в небольшие задачи. По истечении данного периода преподаватель выставляет оценку за выполненное задание всей группе.

Задача относится строго к роли ответственного в проекте. Соответственно, обязанности каждого из студентов зависят от его роли в проекте. Так как реализуемая платформа адаптирована для будущих ИТ-специалистов, основные роли в команде могут быть следующими: руководитель проекта, frontend-разработчик, backend-разработчик и тестировщик. При этом возможны добавление новых ролей или адаптация платформы для специалистов других отраслей. Основными задачами руководителя являются формирование задач из общего задания преподавателя и контроль их выполнения другими участниками. Для других участников главной обязанностью является выполнение строго своих

задач (относящиеся к их роли). Также допускается, что один и тот же студент может быть вовлечен в две разные команды в разных ролях, поэтому необходимо контролировать общую загруженность учащегося.

Деятельность преподавателя на этой платформе можно описать следующим образом: после формирования команд преподаватель назначает задание каждой из групп. Во время его выполнения он может вмешиваться в процесс, помогать, давать советы. Также возможны изменение состава команд и ролей или добавление новых участников. После окончания этого периода команда сдает задание преподавателю и получает общую оценку, которую члены команды делят между собой. Преподаватель может закончить проект, изменить состав команды или оставить его неизменным и добавить новое задание. Далее начинается новый цикл, описанный выше.

Несмотря на полную картину описанного образовательного процесса, он все еще не реализует полностью стратегию STAD. Чтобы достичь этого, необходимо также добавить ту часть стратегии, когда преподаватель обучает студентов и проводит индивидуальное тестирование. Данные требования полностью покрывают большинство современных LMS, в которых имеется функционал для создания и проведения тестов, распространения учебных материалов разного формата. Необходимо лишь провести интеграцию между LMS и платформой.

РЕАЛИЗАЦИЯ

Для реализации необходимых инструментов для поддержки заданий по ролям необходимо определить приоритеты каждого из них. В исследовании, описанном в работе [7], отмечается, что, хотя такие интерактивные инструменты, как чат или вики, высоко оценивались пользователями, согласно логированию использовалось меньше других инструментов. Из этого можно заключить, что приоритетными будут инструменты, связанные с управлением проектом.

Как уже отмечалось ранее, ядром платформы является Gitlab, вокруг которого строится веб-приложение. Оно предоставляет дополнительный функционал для управления проектами и работы по ролям: создание задач, трекинг ошибок, ограничение действий пользователя на Gitlab в зависимости от роли в проекте.

Также имеются инструменты для общения, такие, как чат, заметки внутри проекта, оповещения. Это необходимо для установления контакта, когда члены команды не знакомы друг с другом.

Ключевым функционалом является страница проекта с доской, на которой размещаются задачи. Для каждого периода и каждого члена команды предусмотрен свой раздел. На отдельной странице имеется возможность детального просмотра задачи: прикрепленных документов, полного описания и комментариев.

Основные модули

Серверная часть приложения состоит из следующих блоков:

- Model – базовые сущности, которые отображены в базе данных, скрипты миграции;
- Provider-сервисы, работающие с базой данных. В данном проекте также реализованы провайдеры, контролирующие доступ к данным;
- BackendPortal – веб-приложение, обрабатывающее запросы и хранящее конфигурации;
- PortalService – сервисы и расширения, необходимые для веб-приложения: расширение для случайного выбора из коллекции, сервисы оповещений, работы с файлами, сервис для чата и формирования команд;
- GitlabClient – клиент для работы с Gitlab. Сервисы, формирующие запросы к Gitlab API и десериализующие присланные данные;
- SimulationService – сервисы, отвечающие за симуляции проектов и ролей.

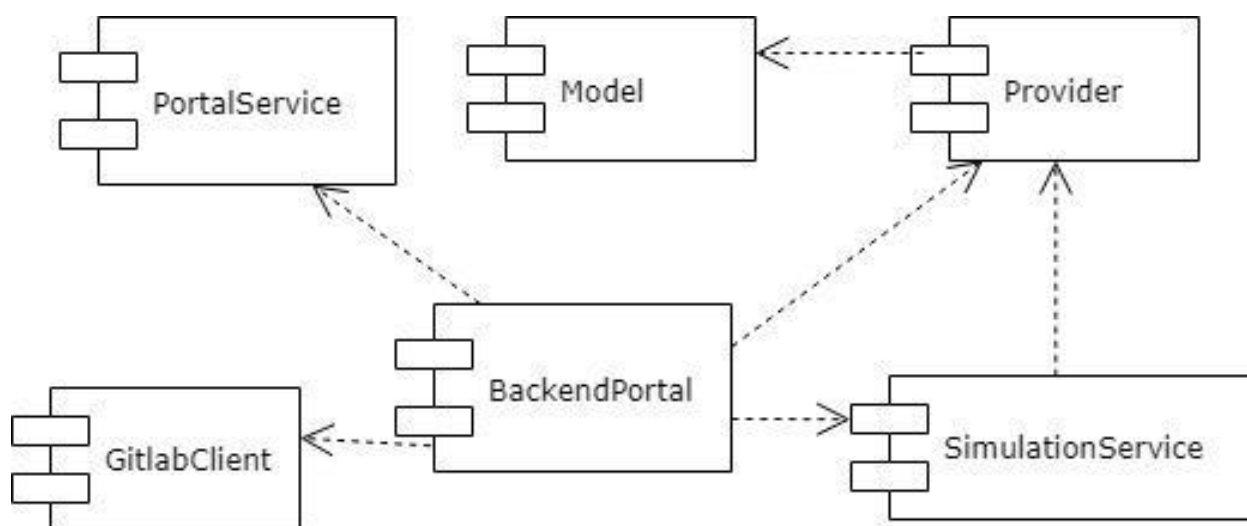


Рис. 1. Диаграмма компонент серверной части приложения

Описание имеющихся сущностей

User – Пользователь системы. Это может быть студент или преподаватель. Каждая из этих ролей имеет свои права на платформе.

Task – Задача на проекте в текущем периоде. Каждая задача прикрепляется только к одному пользователю. Имеет атрибуты приоритета (низкий, средний и высокий), тип (баг или новый функционал), статус (в ожидании, сделана, подтверждена). Задачи могут быть созданы исключительно только преподавателем или руководителем проекта. Каждая задача прикреплена к проекту, пользователю и периоду (Milestone).

Milestone – период, в течение которого группа работает над проектом. Имеет дату начала и конца периода, заголовок и описание. Также у каждого периода есть прикрепленные заметки и чат.

Project – сущность, содержащая данные о проекте: ссылку на gitlab-репозиторий, текущий состав команды с ролями, название, описание и прикрепленные документы.

ProjectRole – роль студента в проекте, например, это может быть разработчик серверной или клиентской части, руководитель проекта или тестировщик.

Безопасность

В качестве основного инструмента для аутентификации и авторизации было решено использовать JWT. JSON Web Token (JWT) – это открытый стандарт

(RFC 7519) для создания токенов доступа, основанный на JSON формате. Как правило, используется для передачи данных авторизации в клиент-серверных приложениях. Токены создаются сервером, подписываются секретным ключом и передаются клиенту, который в дальнейшем использует данный токен для подтверждения своей личности [8].

Провайдеры данных

В качестве базового интерфейса для работы с базой данных был использован CRUD-провайдер с базовыми операциями, реализованными в абстрактном классе:

- найти сущность по уникальному идентификатору;
- найти все сущности из таблицы;
- удалить сущность по уникальному идентификатору;
- создать сущность;
- обновить сущность.

Все остальные провайдеры, наследующие этот класс, дополняли его необходимыми функциями.

Для обеспечения безопасности данных для каждого провайдера был реализован декоратор, реализующий тот же интерфейс и переопределяющий каждый метод. Сначала происходит проверка наличия у пользователя прав на данную операцию. Если прав достаточно, происходит выполнение операции, иначе высылается ошибка клиенту.

Для работы с Gitlab были реализованы сервисы для работы с проектами и пользователями. Например, при создании пользователя в платформе формируется профиль для него на Gitlab, аналогично с проектами. Для доступа был сгенерирован ключ авторизации на Gitlab, который добавляется в заголовки каждого запроса. API данной гит-платформы дает возможность полностью контролировать всеми имеющимися сущностями.

Также разработан сервис для переопределения состава команд: преподаватель может изменить составы команд разных проектов, причем роли и новый состав определит сама платформа. В дальнейшем данный функционал можно будет сделать более адаптивным и формировать новые команды на основе определенных параметров.

Алгоритм формирования команд:

- изначально имеются проекты и позиции, на которые нужно добавить новых студентов;
- также имеются студенты, которых можно использовать в формировании новых команд;
- далее идет итерация по требуемым ролям для каждого проекта; из выбранных студентов берутся те, кто не был еще на этом проекте и не имеет этой роли в другом проекте; из сформированного множества студентов случайным образом выбирается один студент и добавляется к этому проекту.



Рис. 2. Диаграмма симуляции проекта

Сервис оповещений построен следующим образом: все оповещения имеют два статуса (новое оповещение и просмотренное), клиент каждые десять секунд посылает запрос к данному сервису, который высылает новые оповещения. Пользователь может просмотреть оповещение, нажав на специальную кнопку. Тогда в меню на сайте это оповещение будет помечено как прочитанное и далее не будет отображаться в специальном индикаторе новых оповещений.

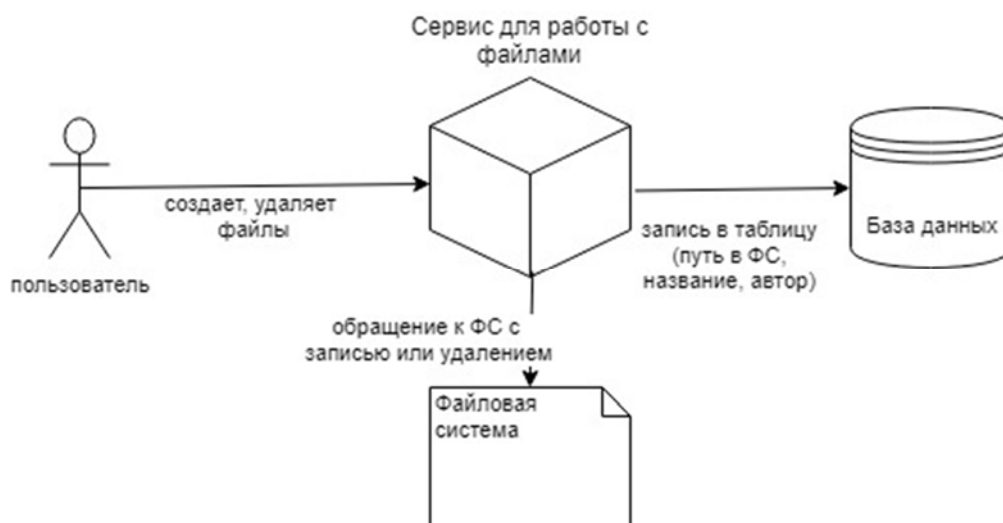


Рис. 3. Сервис для работы с файлами

Для управления файлами разработан сервис, реализующий необходимый функционал, работающий с файловой системой и базой данных. При добавлении файла делается запись в таблице БД и на диске. Все остальные операции происходят тем же образом.

Клиентская часть

Базовой частью всей клиентской части является фреймворк Angular 4 [9]. Была также использована библиотека Angular Material [10], которая позволяет использовать готовые элементы графического интерфейса.

При логине сгенерированный JWT-токен сохраняется в браузере и далее при каждом запросе добавляется в заголовок запроса. При выходе с платформы этот токен удаляется.

ЗАКЛЮЧЕНИЕ

Несмотря на большое количество обучающих платформ разных типов, ни в одной из них не реализована поддержка совместного ролевого выполнения заданий. В данной работе рассмотрен один из возможных вариантов реализации подобной платформы. В качестве основной модели обучения выбрана стратегия STAD кооперативного обучения, а в качестве главного инструмента для поддержки групповой работы над проектами – Gitlab. Реализованная система предоставляет инструменты для управления проектами, поддержки выполнения заданий по ролям, а также имеет дополнительный функционал для обеспечения коммуникаций внутри команд.

В дальнейшем на основе имеющихся данных о проектах и задачах возможно создание симуляций проектов: преподаватель сможет выбрать определенный период реализации проекта и использовать его для обучения новых студентов. Это позволит формировать специальные ситуации для развития определенных компетенций у учащихся.

СПИСОК ЛИТЕРАТУРЫ

1. *Стивен Л., Стефани Д.* Экономия времени или инновационная практика: исследование восприятия и использования систем управления обучением // Эльзевир. 2008. №53 (3). С. 5.
2. *Жао Д., Ксайлонг Ф., Кан Ф., Кифенг Л.* Интерактивная и совместная платформа электронного обучения с интегрированным социальным программным обеспечением и системой управления обучением. Изд-во ИТСЕ, 2012. С. 11–13.
3. *Джон В.* Обзор исследований по обучению на основе проектов. Изд-во The Autodesk Foundation, 2000. С. 1–3.
4. *Джонсон Д., Джонсон Р.* Что заставляет кооперативное обучение работать // Лоуренс Эрлбаум Ассошиэйтс, 1999. №38 (2). С. 2.
5. *Роберт Славин.* URL: <http://education.jhu.edu/directory/robert-slavin-phd/>
6. *Диленбург П.* Что вы подразумеваете под совместным обучением? Изд-во Оксфорд Эльзевир, 2007. С. 4.
7. *Ван Д.Т.* Эффекты стратегии СТАД по академическим достижениям и отношения учащихся средней школы в математике // Международный научный журнал. 2013. С. 1–3.
8. JSON Web Token. URL: https://ru.wikipedia.org/wiki/JSON_Web_Token
9. Angular Framework. URL: <https://angular.io/>
10. Angular Material. URL: <https://material.angular.io/>

INSTRUMENTS SUPPORTING ROLE-BASED EXERCISES USING STAD STRATEGY IN E-LEARNING SYSTEMS

V. V. Matyunin¹, A. A. Marchenko²

^{1,2} Higher School of Information Technologies and Intelligent Systems at Kazan (Volga region) Federal University

¹ vlad.matyunin96@gmail.com, ²anton.a.marchenko@gmail.com

Abstract

In this paper one of the possible implementations of cooperative education models based on STAD (Student Teams-achievement Divisions) strategy of cooperative learning in LMS (Learning Management System) is described. The approaches of this learning methodology increase the level of teamwork skills, need in further professional activity, and their injection into LMS can help to automate and optimize some processes and open new opportunities for implementation of new instruments.

Keywords: cooperative learning, STAD, LMS, e-learning

REFERENCES

1. Steven L., Stephanie D. Saving time or innovating practice: Investigating perceptions and uses of Learning Management Systems // Elsevier. 2008. No 53 (3). S. 5.
2. Zhao D., Xiaolong F., Can F., Qifeng L. Interactive and Collaborative E-Learning Platform with Integrated Social Software and Learning Management System. Izd-vo ITSE, 2012. S. 11–13.
3. John W. A review of research on project-based learning. Izd-vo The Autodesk Foundation, 2000. S. 1–3.
4. Johnson D., Johnson R. What Makes Cooperative Learning Work// Lawrence Erlbaum Associates, 1999. No 38 (2). S. 2.
5. Robert Slavin. URL: <http://education.jhu.edu/directory/robert-e-slavin-phd/>
6. Dillenbourg P. What do you mean by collaborative learning? Izd-vo Oxford Elsevier, 2007. S. 4.

7. *Van D.T.* Effects of Student Teams Achievement Division (STAD) on Academic Achievement, and Attitudes of Grade 9th Secondary School Students towards Mathematics // International Journal of Sciences. 2013. P. 1–3.
8. JSON Web Token. URL: https://ru.wikipedia.org/wiki/JSON_Web-Token
9. Angular Framework. URL: <https://angular.io/>
10. Angular Material. URL: <https://material.angular.io/>

СВЕДЕНИЯ ОБ АВТОРАХ



МАТЮНИН Владислав Владимирович – бакалавр Высшей школы информационных технологий и интеллектуальных систем Казанского (Приволжского) федерального университета.

Vladislav Vladimirovich MATYUNIN – Bachelor of the Higher School of Information Technologies and Intelligent Systems at Kazan (Volga region) Federal University.

e-mail: vlad.matyunin96@gmail.com



МАРЧЕНКО Антон Александрович – преподаватель Высшей школы информационных технологий и интеллектуальных систем Казанского (Приволжского) федерального университета, ассистент кафедры программной инженерии.

Anton Aleksandrovich MARCHENKO – lecturer of the Higher School of Information Technologies and Intelligent Systems at Kazan (Volga region) Federal University, department assistant of software engineering.

e-mail: anton.a.marchenko@gmail.com

Материал поступил в редакцию 2 июня 2018 года