

# Программное обеспечение мониторинга комплекса проектов студентов инженерных специальностей

Е. Н. Десятирикова, О. В. Минакова, А. С. Голиков  
Воронежский государственный технический  
университет  
science2000@ya.ru

Ю. В. Хрипунов  
Орловский государственный университет  
phyrexia@yandex.ru

Л. В. Черненькая  
Санкт-Петербургский политехнический университет Петра Великого  
mv@qmd.spbstu.ru

**Аннотация.** В инженерных дисциплинах наибольшую трудоемкость как для студентов, так и для преподавателей представляет курсовое проектирование. Для удаленного мониторинга проектной деятельности разработан программный инструментарий, проектирование которого описано в данной работе. В основе технологии управления проектной задачей лежит деление на этапы и изменение их статуса по достижению заданных условий. Результаты выполнения курсовых работ состоят из нескольких документов, моделей, программного кода, который необходимо постоянно и поэтапно верифицировать. Управление конфигурацией и отслеживание изменений в артефактах реализовано с использованием автоматического разбора присланных сообщений и последовательной нумерацией оригинальных документов. Взаимодействие между преподавателем и обучающимися организовано на основе почтового ящика. Настройка под конкретную учебную деятельность реализована через добавление и именованние этапов; изменение статуса может осуществляться автоматически и вручную.

**Ключевые слова:** разработка программ, автоматическое оценивание, отслеживание изменений, проектная деятельность

## I. ВВЕДЕНИЕ

Облачные сервисы и большие системы в последнее время не заслуженно вытеснили инструменты индивидуальной работы. Под программным инструментарием понимается десктопное приложение, предназначенное для индивидуальной работы в области профессиональной деятельности. Изначально возникшие в программной инженерии эти программные инструменты оказались эффективны в различных областях творческой и образовательной деятельности [1, 2]. Такие системы должны иметь возможность гибкой и простой настройки под потребности каждого пользователя. При этом они должны следовать стандартам профессиональной деятельности, быть совместимыми с прикладным программным обеспечением, стандартами передачи

данных предметной области и распространенными средствами совместной работы.

Проектная деятельность – это активно развивающийся педагогический подход к организации образовательного процесса [3]. Поэтому в современные образовательные программы школ и университетов по разным направлениям вводят индивидуальные проекты [4]. Кроме того, интенсивно развивается групповая проектная деятельность территориально распределенных обучающихся, дистанционно курируемая Образовательным центром «Сириус». Примером такой деятельности является Всероссийский конкурс научно-технологических проектов старшеклассников и студентов «Большие вызовы», реализуемый в Образовательном центре «Сириус» [5, 6, 7]. В связи с этим необходимы инструменты управления проектами в образовательной сфере.

В отличие от традиционных систем управления проектами, рассчитанных для совместной работы группы над одним проектом, в образовательной деятельности одному куратору необходимо отслеживать изменения или руководить несколькими разными проектами одновременно. При этом специфика курсового проекта по инженерным дисциплинам состоит в том, что создается и формируется из множества проектных артефактов разного вида и формата, которые надо верифицировать между собой для принятия управленческих решений.

Поэтому целью нашей работы являлось построение технологии управления множеством проектов путём контроля последовательности выполнения работ на основе отслеживания изменений в определенной конфигурации артефактов с их одновременной систематизацией.

Результатом этой работы стало создание программного средства для управления курсовым проектированием [8], основанное на принципах системной инженерии, нашем опыте преподавания инженерных дисциплин, методическом руководстве направлением проектирования «Нанотехнологии» Проектных программ «Большие

вызовы» в ОЦ «Сириус», участием в проектно-образовательных интенсивах Университета 2035 и наставничеством в IT-хатаконах.

## II. АНАЛИЗ ПРОБЛЕМНОЙ ОБЛАСТИ

Процесс проведения курсового и дипломного проектирования может быть разбит на ряд последовательных этапов, отличающихся от рекомендованных в своде знаний по управлению проектами РМВОК[9]. Их выбор обусловлен спецификой преподаваемой дисциплины. Например, для программирования на начальных курсах примерное содержание этапов может включать постановку задачи, написание кода программы, оформление пояснительной записки. Для дисциплин старших курсов добавляется деление на проектную (построение UML моделей или моделирование бизнес-процессов) и конструкторскую часть проекта. Поэтому в общем виде рабочий процесс индивидуального проектирования для получения профессиональных компетенций можно представить в виде последовательных этапов и проверяемых результатов каждого из них.

Для контроля выполнения каждого этапа необходимо ввести статусы (рис. 1):

- новый, соответствующий инициализации этапа;
- рабочий, студент работает над проектом, а преподаватель ожидает получения результатов;
- рецензирование, начинается после того, как студент отправил задание текущего этапа на проверку преподавателю и ожидает замечаний и рекомендаций;
- исправление, начинается после уведомления студента о его ошибках и выдачи рекомендаций по их исправлению;
- завершённый.

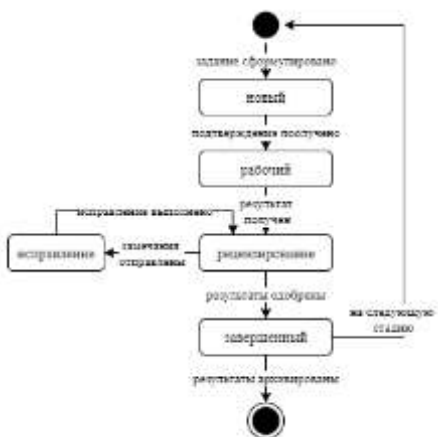


Рис. 1. Диаграмма состояний работы над индивидуальным проектом

Статусы этапов дают возможность оценить активность студента, вовремя активизировать его деятельность, оперативно и автоматически отслеживать работу над

проектом. Такой подход сделает процесс управления курсовым проектированием прозрачным и измеряемым. Для студента эти этапы будут применены в качестве темы сообщений, что облегчит понимание при заочном общении.

## III. ОБОСНОВАНИЕ РЕШЕНИЯ

Так как программный инструмент для управления курсовым проектированием будет строиться на взаимодействии преподавателя со студентом, необходимо определить механизм обмена данными. На сегодняшний день для реализации удаленного взаимодействия необходимо развернуть сервер, который будет принимать результаты проектирования и направлять назад замечания преподавателя. Такой подход требует дополнительных технических ресурсов и трудоемок, поэтому гораздо лучше использовать уже готовый и бесплатный вариант – почтовый сервер. На сегодняшний день существует множество доступных и условно бесплатных сервисов электронной почты, которые успешно применяются для организации безличностного взаимодействия [10].

Преимуществами использования электронного почтового ящика являются оперативность доставки любых сообщений адресату, независимость от местоположения и возможность работы с различных устройств, простота и широкая распространенность среди различных категорий пользователей.

Типичный пользовательский сценарий состоит в том, что преподаватель открывает программу работы с электронной почтой, выбирает письмо, читает сообщение или изучает содержимое вложенного файла, а затем пишет ответное сообщение. В связи с этим возникают случаи, когда студент не добавил файл с результатами, не указал задание или даже прислал анонимное сообщение. Большинство ошибочных ситуаций, которые наставнику приходится обрабатывать вручную могут решаться автоматически.

Другая задача, которую постоянно решает наставник – это верификации программных артефактов. Когда одновременно ведется управление множеством проектов, необходимо отслеживать точную формулировку задания, проверять код, читать пояснительную записку, уточнять обоснование принятых решений через повторную отправку сообщений. При этом каждый элемент проектной конфигурации может находиться в разных сообщениях. Поэтому необходима постоянная трассировка сообщений и результатов по одному проекту в естественной последовательности. Некоторые программы электронной почты имеют такую функцию, но ее корректная работа требует четкой организационной дисциплины участников обмена.

Третья проблема, которая может решаться автоматически – это отслеживание изменений. Автоматическая проверка различий в документах позволит значительно уменьшить трудоемкость работы, и заставит студентов исправлять замечания и присылать оригинальные результаты. Возможности программного средства позволяют вводить разнообразные условия и

критерии анализа присланных результатов. В практике обучения программной инженерии накоплен большой опыт использования методов автоматического динамического и статистического тестирования программ, а также разработано и активно развивается разнообразное ПО [11]. Но такие инструменты в отличие от предлагаемого универсального решения специфичны.

Предлагаемый сценарий автоматизации отслеживания изменений в работе над проектом представлен на рис. 2. Студент будет отправлять преподавателю на его почтовый ящик письма/файлы, предназначенные для обсуждения курсового проекта. Преподаватель с помощью программного инструмента сможет проверить всю почту на наличие писем, связанных с курсовым проектированием, распределит эти письма по отдельным, выделенным для каждого студента папкам, и сохранит всю историю изменений и отправленных сообщений. Преподаватель, проверив работу, сможет отправить заключение или рекомендации по присланной студентом работе.



Рис. 2. Сценарий удаленного взаимодействия по управлению заданиями

Разработанный программный продукт работает независимо и может быть развернут у любого пользователя с активным почтовым ящиком. Наше обоснование выбора почтового сервиса для этой задачи представлено в [12] и во многом зависит от национальных правил.

Проблема определения набора проектных артефактов решается для каждой дисциплины и проекта отдельно. В стандартном варианте предполагается следующая конфигурация:

- техническое задание на проектирование в виде текстового документа;
- код программы в файлах с расширением имен компилятора языка программирования;
- пояснительная записка в формате, совместимом с Word.

В качестве универсализации подхода мы предлагаем установку количества и названий папок для хранения результатов, а также введения средств автоматической очистки мусора. Так по завершению каждого этапа перед установкой статуса «завершено» система будет предлагать удалить все версии документов, кроме последней, относящиеся к текущему этапу.

После завершения проекта будет выполняться архивирование проекта по заранее определенному

шаблону выбора содержимого из папок и формирования идентификатора архива.

#### IV. ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО РЕШЕНИЯ

В качестве архитектурного стиля данного решения настраивается архитектура клиент-сервер. Но в роли сервера, выступает общедоступный почтовый интернет сервер, принимающий письма с вложениями от студентов и перенаправляющий ему ответы из разработанного программного инструмента. Поэтому в реализации сервера нет необходимости. Модульное представление клиентской части программного средства показано на рис. 3.

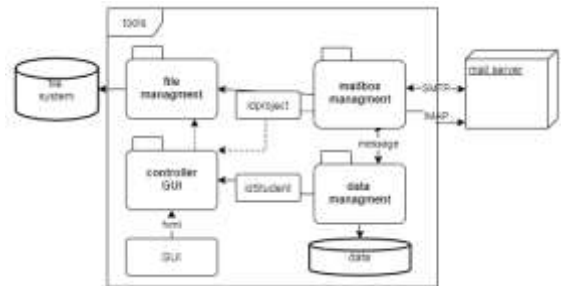


Рис. 3. Модульная декомпозиция ПС

Взаимодействие студента/преподавателя с приложением осуществлено посредством интерфейса пользователя (GUI) на платформе JavaFx. Выбор платформы обусловлен нашим предпочтением Java в качестве языка разработки. Помимо обычных элементов управления она предоставляет готовый компонент web-view, который можно настроить как собственный браузер, и средства работы с мультимедиа, что может быть использовано для наращивания функциональности.

Для управления студентами потребовалась небольшая база данных, которая будет хранить идентификационную информацию, почтовый адрес студента и текущее состояние проекта, включая статусы этапов. В качестве БД была выбрана SQLite, так как эту базу данных можно подключить и включить в проект, в то время как другие требуют настройки и развертывания сервера СУБД. Для использования SQLite необходимо установить дополнение sqlite-jdbc. После установки можно использовать драйвер JDBC для SQLite с помощью библиотеки org.sqlite.JDBC. В реализации был использован паттерн data access object (DAO), что позволило вынести функции настройки данных приложения в интерфейс пользователя. Также это позволит использовать любую БД для хранения состояния проектов. Модель данных приложения представлена на рис. 4.

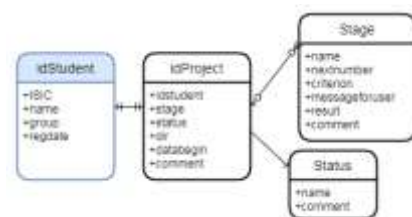


Рис. 4. Модель данных

С позиции интероперабельности с LMS необходимо было исключить идентификационную информацию студента из модели данных, но это потребовало дублирования слоя DAO в контролере GUI. Путь обезличенного общения с подстановкой имени студента автоматически в модели почтового аккаунта на наш взгляд не приемлем в образовательной деятельности. В результате сущность idStudent оказалась частично продублирована в idProject, путем введения атрибутов для хранения почтового адреса и имени студента, также избыточно фиксируется и дата начала проекта.

Модуль управления почтовым аккаунтом (mail box management) состоит из SMTP и IMAP модулей, которые осуществляют отправку запросов на сервер по соответствующим протоколам на сервер Yandex.ru, а также логику для разбора полученных сообщений, их сортировки по темам.

Для начала работы преподавателю необходимо осуществить вход в почтовый аккаунт, после этого все стандартные опции работы с почтой – просмотр входящих прочитанных и непрочитанных сообщений, смена статуса сообщения (прочитано/непрочитано), скачивание вложений и написание ответа станут доступны. Чтобы облегчить БД было принято решение не хранить в ней историю сообщений электронной почты, а восстанавливать ее из цепочки писем по теме сообщений.

Все связывание между письмами и хранимыми файлами основано на сущности Stage. Ее имя используется и для заголовков писем, и для папок хранения результатов, таким образом, никаких дополнительных связываний по данным между программными пакетами не требуется. Атрибут dir хранит имя служебной директории для хранения данных всей программы и должен быть создан в начале работы после загрузки данных о студентах. Далее размещение полученных от студентов материалов осуществляется по шаблону, где ver – присваиваемый номер версии:

`dir/nickname/stage/result<ver>`,

Модуль управления файлами (file management) формирует этот путь по представленному выше шаблону, добавляя следующий порядковый номер версии по типу содержимого определенного атрибутом result. Также автоматически происходит удаление предыдущих версий и архивирование старшей при переходе проекта в статус завершено.

Гибкость настройки рабочего процесса управления студенческой работой обеспечивается за счет возможности задания произвольного количества стадий и статусов проекта. Для каждой стадии предусмотрено задание имени, ссылки на следующую, критерия перехода, а также включение длинного описания в виде комментария и сообщения для студента. Названия статусов и комментариев к ним также могут быть переопределены пользователем при инициализации проекта. Если предложенная схема переходов (рис. 1) не устраивает преподавателя, он может изменить статус вручную, выбрав его из списка в рабочем окне программы. Таким образом, разработанное программное средство позволяет автоматизировать управление, при этом сохраняя возможность гибкой настройки путем включения дополнительных этапов.

## V. ЗАКЛЮЧЕНИЕ

Целесообразность автоматизации процесса управления курсовым проектированием обусловлена необходимостью отслеживания изменений в нескольких наборах проектных артефактов одновременно. Проведенный анализ показал, что имеющиеся системы управления проектами предназначены для совместной работы над одним проектом, требуют опыта использования и не могут быть настроены под индивидуальную работу. На основе проведенного моделирования предметной области было составлено обобщенное описание рабочего процесса управления курсовым проектированием, как последовательность этапов и их статусов. Клиент-серверная архитектура была положена в основу проектного решения. Для реализации хранения данных была спроектирована легкая БД {студенты, этапы, статусы, каталог артефактов}, процесс обмена сообщениями со студентами организован через почтовый сервис Yandex, что позволит облегчить установку и дальнейшую эксплуатацию решения.

Разработанное ПС загружено на github (<https://github.com/SashaSunshine/CourseProjectsManager>) и может быть использовано «как есть» или доработано по лицензии MIT.

## СПИСОК ЛИТЕРАТУРЫ

- [1] Fowler L., Armarego J., Allen M. Case tools: Constructivism and its application to learning and usability of software engineering tools // Computer Science Education. 2001. Vol. 11. N. 3. pp. 261-272.
- [2] Starodubtsev I.Yu., Desyatnikov F.A., Nuzhnykh A.V. "Android Software for Testing in Audio and 2D User Interaction Environment", 2020 IEEE International Conference Quality Management, Transport and Information Security, Information Technologies (IT&QM&IS), 7-11 Sept. 2020, Yaroslavl, Russia DOI: 10.1109/ITQMIS1053.2020.9322921
- [3] Bell S. Project-based learning for the 21st century: Skills for the future // The clearing house. 2010. Vol. 83. N. 2. pp. 39-43.
- [4] Pak V.V. Project-based learning as a method of formation of universal project skills of engineering university students // Pedagogical education in Russia, vol. 1, pp. 68-74, 2016.
- [5] Desyatnikov F.A., Khripunov Yu.V., Gribanov E.N. Android application for a sensory test system for the determination of tetracycline antibiotics // 2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), St. Petersburg, Moscow, Russia, 2021, DOI: 10.1109/EIConRus51938.2021.9396202/
- [6] Desyatnikov F.A., Danilova A.V., Gubkin I.M. Biomedical Applications of Nanotubes in Doping Problems // 2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (2019 EIConRus), St. Petersburg and Moscow, Russia, January 28-30, 2019. DOI: 10.1109/EIConRus.2019.8656743.
- [7] Desyatnikov F.A., Khripunov Yu.V. Metalens Concept for Focusing Wi-Fi Radiation // 2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), St. Petersburg, Moscow, Russia, 2020, Doi: 10.1109/EIConRus49466.2020.9039010.
- [8] Минакова О.В., Голиков А.С. Свидетельство о государственной регистрации программы для ЭВМ №2021616355 20.04.2021г. «Цифровой помощник руководителя курсового проектирования».
- [9] A Guide to the Project Management Body of Knowledge (PMBOK Guide) Fifth Edition. Retrieved 2021-06-22.
- [10] Cao J. et al. Mailbox-based scheme for mobile agent communications // Computer. 2002. Vol. 35. N. 9. pp. 54-60.
- [11] Gordillo A. Effect of an instructor-centered tool for automatic assessment of programming assignments on students' perceptions and performance // Sustainability. 2019. T. 11. №. 20. С. 5568.
- [12] Голиков А.С., Минакова О.В. Создание программного средства для управления курсовым проектированием инженерных дисциплин // Информационные технологии в строительных, социальных и экономических системах. 2021. № 1 (23). С. 107-111.