УДК 004.42:378

ДИНАМИЧЕСКАЯ АКТУАЛИЗАЦИЯ РПД НА ОСНОВЕ ИНТЕГРАЦИИ ГЕТЕРОГЕННЫХ ДАННЫХ

Кулик Иван Вадимович[1], Беднякова Татьяна Михайловна[2], Сычева Маргарита Петровна[3], Русакова Елена Александровна[4]

[1]Студент;
Государственный университет «Дубна»;
Россия, 141980, Московская обл., г. Дубна, ул. Университетская, 19;
e-mail: kivo.22@uni-dubna.ru.

[2]Старший преподаватель;
Государственный университет «Дубна»;
Россия, 141980, Московская обл., г. Дубна, ул. Университетская, 19;
e-mail: tmbednyakova@uni-dubna.ru.

[3]Старший преподаватель;
Государственный университет «Дубна»;
Россия, 141980, Московская обл., г. Дубна, ул. Университетская, 19;
e-mail: msytcheva@yandex.ru.

[4]Старший преподаватель;
Государственный университет «Дубна»;
Россия, 141980, Московская обл., г. Дубна, ул. Университетская, 19;
e-mail: rusakova@uni-dubna.ru.

В статье рассматривается подсистема динамической актуализации рабочих программ дисциплин (РПД) для ИС «Конструктор РПД» Университета «Дубна». В отличие от аналогов, автоматизирующих лишь первичное создание РПД, подсистема обеспечивает их динамическую синхронизацию с учебными планами и матрицами компетенций. Преподаватель при этом работает исключительно с содержательной частью программы. Жизненный цикл документа контролируется через конечно-автоматную модель с фиксацией истории переходов. Это снижает трудоемкость и исключает ошибки ручного переноса данных. Архитектура реализована на стеке React, Node.js/Express и PostgreSQL (JSONB). Эксперимент на 63 дисциплинах подтвердил сокращение времени актуализации комплекта с 20 часов до 14 минут (≈ 99%) при полном устранении ошибок переноса. Подсистема прошла апробацию в Университете «Дубна».

Ключевые слова: рабочая программа дисциплины (РПД), автоматизация документооборота, динамическая актуализация, интеграция данных, метод глубокого слияния JSONB, учебный план, матрица компетенций, информационная система вуза.

Для цитирования:

Динамическая актуализация РПД на основе интеграции гетерогенных данных / И. В. Кулик, Т. М. Беднякова, М. П. Сычева, Е. А. Русакова // Системный анализ в науке и образовании. 2026. № 3. С. 48–57. EDN: EWLITY. URL: https://sanse.ru/index.php/sanse/article/view/735.

DYNAMIC UPDATING OF ACADEMIC DISCIPLINE PROGRAMS BASED ON THE INTEGRATION OF HETEROGENEOUS DATA

Kulik Ivan V.[1], Bednyakova Tatyana M.[2], Sycheva Margarita P.[3], Rusakova Elena A.[4]

Статья находится в открытом доступе и распространяется в соответствии с лицензией Creative Commons «Attribution» («Атрибуция») 4.0 Всемирная (CC BY 4.0) https://creativecommons.org/licenses/by/4.0/deed.ru

[1]*Student;*
Dubna State University;
19 Universitetskaya st., Dubna, Moscow region, 141980, Russia;
e-mail: kivo.22@uni-dubna.ru.

[2]*Senior teacher;*
Dubna State University;
19 Universitetskaya st., Dubna, Moscow region, 141980, Russia;
e-mail: tmbednyakova@uni-dubna.ru.

[3]*Senior teacher;*
Dubna State University;
19 Universitetskaya st., Dubna, Moscow region, 141980, Russia;
e-mail: msytcheva@yandex.ru.

[4]*Senior teacher;*
Dubna State University;
19 Universitetskaya st., Dubna, Moscow region, 141980, Russia;
e-mail: rusakova@uni-dubna.ru.

The paper addresses a subsystem for the dynamic updating of course syllabi (rabochaya programma distsipliny, RPD) within the "RPD Constructor" information system of Dubna University. Unlike existing solutions, which automate only the initial creation of syllabi, the proposed subsystem provides their dynamic synchronization with curricula and competency matrices, while the instructor works exclusively with the content component of the document. The document's life cycle is governed by a finite-state automaton model that records the full history of state transitions. This reduces labor input and eliminates errors arising from manual data transfer. The architecture is built on a React, Node.js/Express, and PostgreSQL (JSONB) stack. An experiment covering 63 disciplines confirmed a reduction in the time required to update a complete set of syllabi from 6 hours to 14 minutes (≈98%), with data-transfer errors completely eliminated. The subsystem has been piloted at Dubna University.

Keywords: course syllabus (RPD), document workflow automation, dynamic updating, data integration, *JSONB deep-merge method, curriculum, competency matrix, university information system.*

For citation:

Kulik I. V., et al. Dynamic updating of academic discipline programs based on the integration of heterogeneous data. *System analysis in science and education*, 2026;(3):48–57 (in Russ). EDN: EWLITY. Available from: https://sanse.ru/index.php/sanse/article/view/735.

Введение

Рабочая программа дисциплины (РПД) является ключевым нормативно-методическим документом, регламентирующим содержание и оценку качества освоения дисциплины. Необходимость ежегодной актуализации РПД обусловлена регулярной корректировкой учебных планов, обновлением нормативно-правовой базы высшего образования и локальных актов вуза, а также изменением требований профессиональных стандартов и рынка труда.

В большинстве ВУЗов формирование комплекта РПД реализуется через специализированные ИС [3, 4]. Так, платформа «1С:Университет ПРОФ» с расширением «Интеллект Инфо: Образовательные программы» автоматизирует перенос данных учебного плана в рабочие программы [2]; сервис «Конструктор образовательных программ» Университета ИТМО обеспечивает формирование и верификацию РПД [6]; а АИС «Деканат» решает смежные задачи учебно-методического документооборота вуза [8]. В университете «Дубна» используется веб-приложение «Методический конструктор» (далее – ИС «Конструктор РПД») [1].

Перечисленные ИС решают задачу первичного формирования РПД, но не обеспечивают их последующую актуализацию при изменении внешних условий [7]. Это приводит к рассогласованию данных между учебным планом, рабочими программами дисциплин, фондами оценочных средств

(ФОС) и библиотечными ресурсами, что снижает качество управления образовательным процессом [5].

Цель работы – разработка и апробация подсистемы актуализации РПД, обеспечивающей синхронизацию данных с внешними источниками и минимизацию ручного труда.

Критерий оценки результата: автоматизация актуализации РПД на основе интеграции данных из внешних источников позволяет сократить временные затраты не менее чем на 90% и полностью исключить ошибки ручного переноса данных.

Методы исследования: системный анализ, хронометраж процессов, прототипирование, экспериментальная оценка эффективности.

1. Анализ процесса актуализации РПД

1.1 Структура и источники данных РПД

Анализ структуры РПД позволяет выделить два контура данных.

Инвариантный контур содержит нормативные данные из учебного плана, импортируемые из «1С:Университет»: код и наименование дисциплины, семестр, трудоёмкость в зачётных единицах (ЗЕТ), распределение часов по видам занятий, формы контроля, компетенции и индикаторы. Эти данные недоступны для редактирования внутри отдельной дисциплины.

Вариативный контур представляет собой содержательно-методический блок, формируемый преподавателем: темы и разделы дисциплины, распределение часов, методические указания, фонд оценочных средств.

Под комплектом далее понимается совокупность РПД всех дисциплин одного профиля подготовки на учебный год. Матрица компетенций формируется в программе «Планы» и загружается в подсистему файлом (*.xlsx/.csv*).

1.2 Анализ существующего процесса (AS-IS)

Для оценки трудоёмкости и выявления источников ошибок проведён хронометраж процесса актуализации РПД в университете «Дубна». Наблюдения охватывали работу руководителей образовательных программ (РОП), преподавателей и проверяющих при подготовке комплекта для направления подготовки «Информатика и вычислительная техника» (63 дисциплины). Для каждого этапа рассчитано среднее значение и среднеквадратичное отклонение.

Табл. 1. Хронометраж операций при актуализации одной РПД

Операция	Время, мин (среднее)	Средне-квадратичное отклонение	Кол-во замеров
Сверка данных матрицы компетенций и РПД	3	0,8	63
Сверка данных учебного плана и РПД	3	1,2	63
Внесение изменений, редактирование разделов (рассматриваются только изменения в инвариантном контуре)	14	11,6	14

Всего:	20		

Проверка комплекта из 63 дисциплин занимает не менее 6 часов, а полная актуализация — около 21 часа.

Системный источник ошибок – ручное копирование данных из учебного плана (семестр, трудоёмкость в ЗЕТ, распределение часов по видам занятий) и матрицы компетенций (расхождения в компетенциях и индикаторах). Такие опечатки и пропуски приводят к рассогласованию документа с нормативными требованиями.

Полученные результаты обосновывают требования к подсистеме (раздел 2) и проектные решения по её реализации (раздел 3).

2. Требования к подсистеме актуализации

2.1 Функциональные требования

1. Подсистема должна обеспечивать автоматическое извлечение и обновление нормативных данных из «1С:Университет» (код, наименование, семестр, трудоёмкость в ЗЕТ, распределение часов по видам занятий, формы контроля), а также матрицы компетенций без возможности ручного редактирования этих полей в карточке РПД. При обнаружении расхождений между текущими данными РПД и обновлённым учебным планом подсистема должна отображать пользователю разницу до подтверждения применения изменений.
2. Подсистема должна предоставлять преподавателю возможность формировать и редактировать темы/разделы дисциплины, методические указания и фонд оценочных средств (вариативная часть) без влияния на защищённые инвариантные поля, а также обеспечивать повторное использование содержательных материалов из РПД других лет, направлений и форм обучения.
3. Подсистема должна поддерживать статусную модель жизненного цикла РПД («Актуализирована» → «Отправлена преподавателю» → «В работе» → «Готова» → «На проверке» → «На доработке» → «Утверждена») с фиксацией истории переходов и ограничением действий пользователей в зависимости от роли.
4. Подсистема должна реализовывать возможность оставлять структурированные комментарии к конкретным элементам вариативной части с отправкой уведомления. Комментарии проверяющего должны автоматически переводить РПД в статус «На доработке», наличие открытых комментариев должно блокировать утверждение документа.

2.2 Нефункциональные требования

1. Производительность: основные операции (получение списка изменений, сохранение раздела, формирование документа) должны выполняться за время, приемлемое для интерактивной работы, при типовом объёме комплекта (до 65 дисциплин).
2. Отказоустойчивость: при недоступности внешних источников подсистема должна сохранять ранее загруженные данные и информировать пользователя об ошибке.
3. Безопасность: доступ к комплектам и действиям должен определяться ролью пользователя.

2.3 Ролевая модель

В подсистеме выделены три роли пользователей.

РОП (руководитель образовательной программы) – инициирует актуализацию комплекта: обновляет нормативные данные из «1С:Университет», загружает матрицу компетенций, назначает преподавателей, ответственных за содержание РПД. Контролирует прохождение всех этапов.

Преподаватель – редактирует содержательные разделы РПД в рамках вариативного контура: заполняет темы, методические указания, фонд оценочных средств, использует импорт материалов из РПД других лет.

Проверяющий – просматривает РПД, анализирует полноту и корректность содержательных разделов, оставляет комментарии к конкретным элементам вариативной части.

2.4 Модель жизненного цикла РПД

В таблице 2 представлен жизненный цикл РПД с указанием инициатора и условий выполнения.

Табл. 2. Переходы между статусами РПД и их инициаторы

Статус	Инициатор	Событие/ Примечание
«Актуализирована»	РОП	Первичная загрузка или обновление данных учебного плана, загрузка матрицы компетенций/После синхронизации с учебным планом или загрузки матрицы компетенций
«Отправлена преподавателю»	РОП	Отправка РПД преподавателю / После назначения преподавателя на дисциплину и отправки ему РПД для редактирования
«В работе»	Преподаватель	Открытие РПД преподавателем / Начало редактирования РПД преподавателем
«Готова»	Преподаватель	Отправка на проверку / Завершение редактирования или устранение замечаний, отправка РПД на проверку
«На проверке»	Проверяющий	Открытие РПД проверяющим / Когда РПД готова для утверждения и открыта для анализа проверяющим
«На доработке»	Проверяющий или РОП	Добавлен комментарий / При создании комментария с замечаниями
«Утверждена»	Проверяющий	Утверждение документа / При отсутствии замечаний проверяющий утверждает итоговый документ

Графическое представление модели жизненного цикла в виде диаграммы состояний приведено на рис. 1.

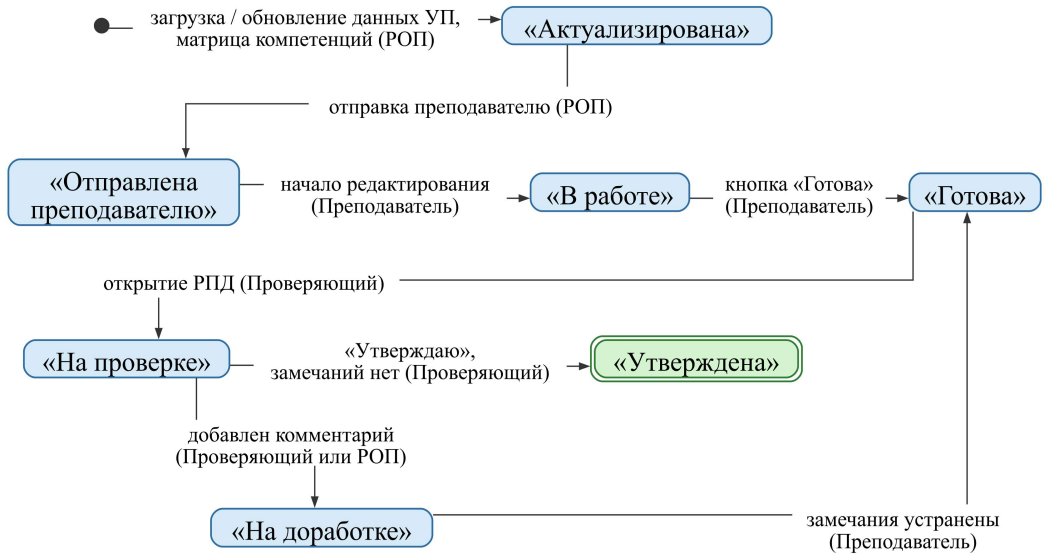


Рис. 1. Диаграмма состояний РПД

Статус «Актуализирована» является доступным для перехода из любого другого состояния модели. Переходы, не указанные в диаграмме, считаются недопустимыми и блокируются серверным слоем подсистемы.

3. Архитектура и реализация подсистемы

3.1 Общая архитектура

Подсистема актуализации построена как расширение существующей ИС «Конструктор РПД» по клиент-серверной архитектуре (рис. 2).

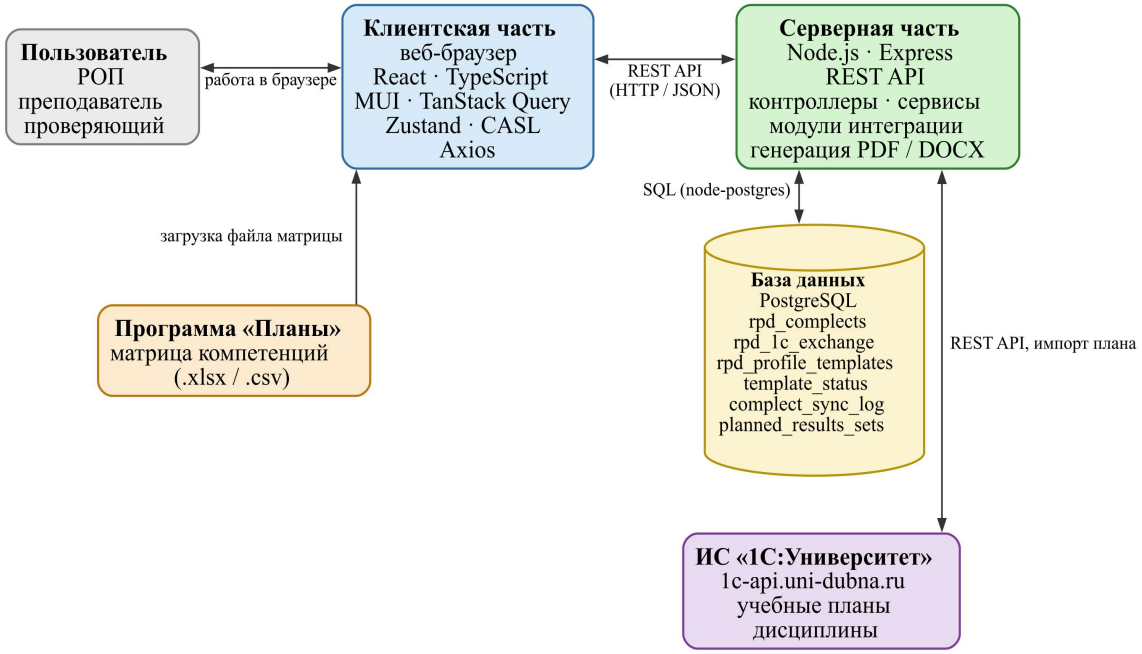


Рис. 2. Общая схема взаимодействия клиента, сервера, базы данных и внешних источников

Клиентская часть реализована на *React* с использованием *TypeScript*; серверная — на *Node.js* с фреймворком *Express.js*; в качестве СУБД используется *PostgreSQL*.

Выбор технологического стека обусловлен:

- требованиями масштабируемости и производительности;
- необходимостью интеграции с существующей системой без переработки архитектуры;
- поддержкой полуструктурированных данных через *JSONB*.

3.2 Модель данных с использованием JSONB

Ключевой архитектурной особенностью является использование полей типа *JSONB* в *PostgreSQL* для хранения гибких, полуструктурированных разделов РПД. Это позволяет:

1. Хранить сложные вложенные структуры (содержание дисциплины, ФОС, историю изменений) без изменения схемы БД.
2. Поддерживать индексирование (*GIN*) и эффективный поиск внутри *JSON*-структур.
3. Сохранять обратную совместимость при добавлении новых полей.

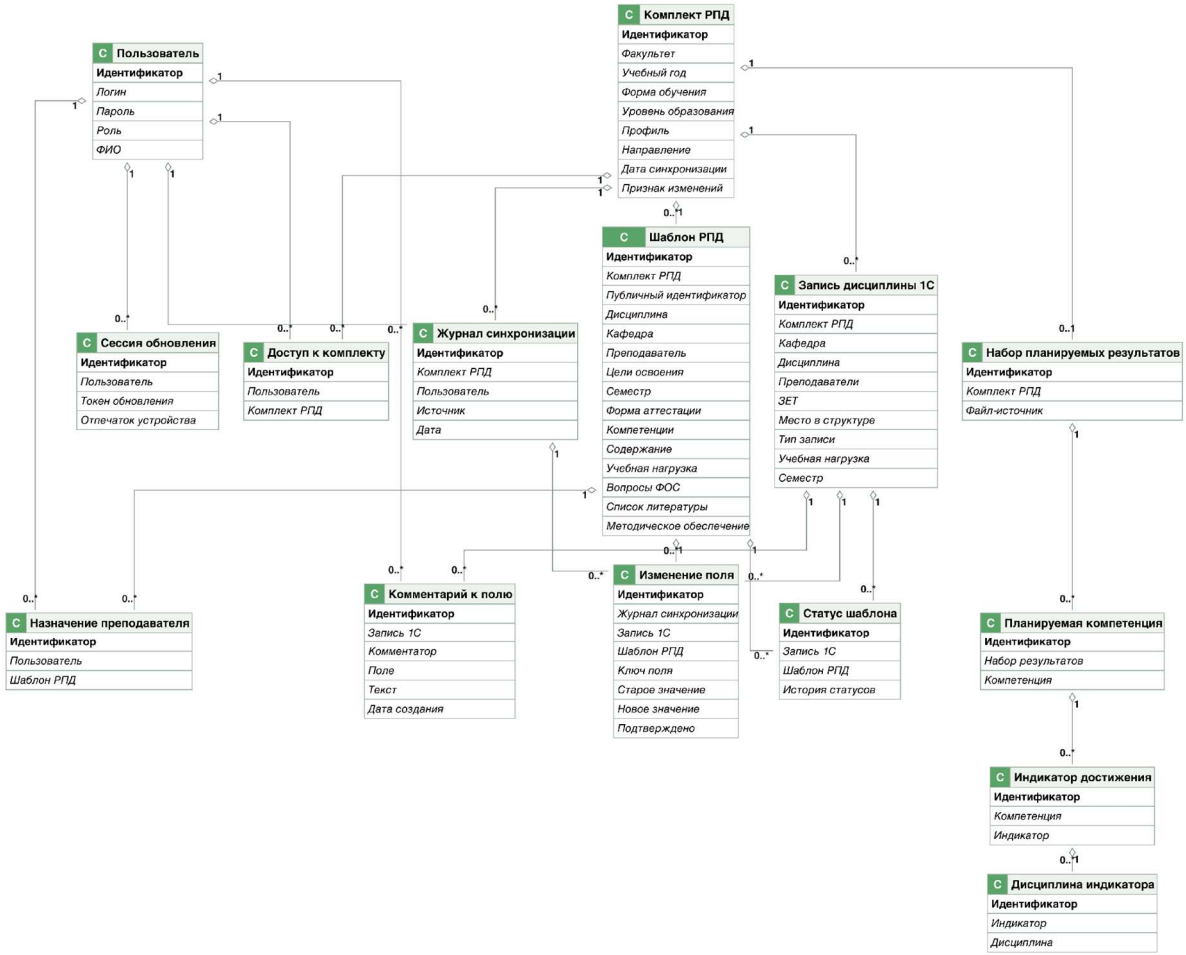


Рис. 3. Логическая модель данных подсистемы актуализации РПД

3.3 Архитектура интеграции с внешними источниками

Источники данных гетерогенны: они различаются протоколом (*REST API* и файловая загрузка), форматом (*JSON* и *.xlsx/.csv*) и режимом обмена (синхронный по запросу и асинхронный).

3.3.1 Интеграция с «1С:Университет» (синхронная, по запросу)

Данные учебного плана извлекаются через *REST API*, предоставляемый внешней системой. Серверный модуль выполняет *HTTP*-запросы с аутентификацией по *API*-ключу. Формат обмена — *JSON*. Запрос инициируется пользователем (РОП) через интерфейс подсистемы и выполняется синхронно: после получения данных сервер формирует дельту изменений и возвращает её клиенту для предварительного просмотра (см. раздел 3.4.1).

При недоступности внешнего *API* система использует кешированные данные, сохранённые при последнем успешном запросе, и отображает пользователю предупреждение о невозможности получения актуальной версии.

3.3.2 Интеграция с программой «Планы» (асинхронная, файловая)

Матрица компетенций загружается из файлов *.xlsx* или *.csv*. Выбор файлового формата обусловлен тем, что программа «Планы» не предоставляет программного *API*, а штатным способом экспорта данных является выгрузка в табличном формате. Парсинг файла выполняется на клиентской стороне с использованием библиотеки *ExcelJS.*

Сформированная *JSON*-иерархия передаётся на сервер (см. раздел 3.4.2).

При ошибке формата или структуры файла пользователь уведомляется о характере ошибки; данные не передаются на сервер, что гарантирует целостность матрицы компетенций.

3.4 Реализация ключевых сценариев

3.4.1 Синхронизация с «1С:Университет» на основе сравнения версий

Полученный по *REST API JSON*-объект учебного плана поэлементно сравнивается с текущими данными; формируется дельта, классифицирующая изменения по типам: новые дисциплины, измененные параметры, исключенные дисциплины. Применение изменений выполняется только после подтверждения пользователем. Это исключает ошибочную автоматическую перезапись данных и дает РОП возможность оценить масштаб изменений. Применение дельты затрагивает только поля инвариантного контура; вариативная часть, заполненная преподавателем, при синхронизации не изменяется. Фиксация изменений осуществляется в рамках единой транзакции, гарантирующей согласованность данных.

3.4.2 Обновление матрицы компетенций в рамках единой транзакции

Загрузка матрицы компетенций реализована по схеме атомарного обновления. Парсинг на клиентской стороне снижает нагрузку на сервер и обеспечивает мгновенную валидацию структуры. Сформированная *JSON*-иерархия {компетенция → индикатор → список дисциплин} передается на сервер, где выполняется единая транзакция: удаление всех записей матрицы для данного года набора и запись новых. При ошибке на любом этапе транзакция отменяется, система возвращается к предыдущему состоянию. После успешного сохранения подсистема автоматически пересчитывает связи компетенций для всех дисциплин комплекта. Фонд оценочных средств дисциплины переносится на новые компетенции и индикаторы, после предварительного согласования.

3.4.3 Импорт содержания из РПД прошлых лет на основе слияния JSONB-структур

Для повторного использования содержательных материалов предложен метод глубокого слияния *JSONB*-структур. В зависимости от раздела РПД, импортируемые данные или объединяются с уже существующими (сюда относятся данные, добавляемые через поля ввода) или заменяются целиком (это структурные *JSONB*-поля: тематический план, фонд оценочных средств). Ключевая особенность – нормативные данные не копируются, что предотвращает внесение устаревших сведений. Реализация на уровне *PostgreSQL* с использованием операторов *JSONB* позволяет выполнять слияние эффективно, без извлечения данных в прикладной код.

3.4.4 Модель управления жизненным циклом РПД на основе конечного автомата

Управление жизненным циклом РПД реализовано по модели детерминированного конечного автомата, формально задаваемого:

- множеством состояний Q = {«Актуализирована», «Отправлена преподавателю», «В работе», «Готова», «На проверке», «На доработке», «Утверждена»},
- множеством событий Σ = {данные_обновлены, отправить_преподавателю, начать_редактирование, завершить_редактирование, отправить_на_проверку, утвердить, создать_комментарий, устранить_комментарии}
- функцией переходов δ: $Q \times \Sigma \rightarrow Q$.
- Примеры допустимых переходов:
- δ(«Актуализирована», отправить_преподавателю) = «Отправлена преподавателю»;
- δ(«Готова», отправить_на_проверку) = «На проверке»;
- δ(«На проверке», создать_комментарий) = «На доработке»;
- δ(«На доработке», устранить_комментарии) = «Готова».

Если событие не соответствует ни одному допустимому переходу из текущего состояния (например, попытка утвердить РПД из состояния «В работе») или нарушено условие, серверный слой блокирует переход и возвращает пользователю сообщение об ошибке.

При успешном переходе выполняется запись в журнал аудита содержащая дату, новое состояние, инициатора и комментарий.

4. Оценка эффективности

В таблице 3 приведено сравнение временных затрат на актуализацию полного комплекта из 63 РПД.

Табл. 3. Сравнение временных затрат на актуализацию комплекта из 63 РПД

Операция	Время, мин (среднее)	Время с подсистемой, мин
Сверка данных из РПД и матрицы компетенций	189	1
Сверка данных учебного плана с РПД (без исправления)	189	5
Внесение изменений, редактирование разделов (рассматриваются только изменения в инвариантном контуре)	882	8
Всего:	1260 (01 час)	14

Сокращение достигается заменой ручной сверки автоматическим построением дельты, а ручного переноса значений – транзакционным применением подтверждённой дельты ко всем РПД комплекта; остаточное время составляют просмотр и подтверждение дельты пользователем.

Заключение

В ходе исследования решена актуальная научно-практическая задача создания подсистемы динамической актуализации рабочих программ дисциплин, функционирующей в условиях гетерогенных источников данных. Основные результаты:

Разработана и апробирована архитектура интеграции, обеспечивающая согласованное обновление данных из двух внешних источников (учебный план «1С:Университет» и матрица компетенций программы «Планы») без потери пользовательских правок.

Предложен и реализован метод детекции изменений в учебных планах с визуализацией дельты, позволяющий осуществлять контролируемое обновление данных на уровне комплекта дисциплин.

Разработана статусная модель и алгоритмы управления жизненным циклом РПД, гарантирующие прослеживаемость всех этапов подготовки документа.

Практическая значимость работы подтверждена опытной эксплуатацией разработанной подсистемы в университете «Дубна». Временные затраты на актуализацию комплекта сокращены примерно на 99%; полностью устранены ошибки, характерные для ручного ввода. Ограничением решения является привязка интеграционных модулей к формату *REST API* «1С:Университет» и структуре выгрузки программы «Планы»: их изменение требует адаптации модулей-адаптеров без изменения ядра подсистемы.

Список источников

1. Беднякова Т. М., Миловидова А. А., Русакова Е. А. Автоматизация процесса подготовки комплекта рабочих программ дисциплин (модулей) в рамках цифровой трансформации университета «Дубна» // Системный анализ в науке и образовании. 2023. № 1. С. 142–150.
2. Майер С. Ф., Муратова Г. В. Анализ систем автоматизации формирования рабочих программ учебных дисциплин // Инженерный вестник Дона. 2021. № 4. С.274-287.
3. Космачёва И. М., Квятковская И. Ю., Сибикина И. В. Автоматизированная система формирования рабочих программ учебных дисциплин // Вестник Астраханского государственного технического университета. Серия: Управление, вычислительная техника и информатика. – 2016. – № 1. – С. 90-97.
4. Кряжев С. А., Левковский Д. И. Информационная система поддержки рабочей программы дисциплины // Вестник Южно-Уральского государственного университета. Серия: Компьютерные технологии, управление, радиоэлектроника. – 2019. – Т. 19, № 3. – С. 135-146. – https://doi.org/10.14529/ctcr190313.
5. Самохина В. М. Автоматизация учебно-методической документации образовательной организации // Современное педагогическое образование. – 2023. – № 10. – С. 311-314.
6. Алгоритмы экспертизы и верификации рабочих программ дисциплин для использования в информационных системах / А. И. Говоров, К. П. Бабаянц, М. М. Говорова [и др.] // Вестник Астраханского государственного технического университета. Серия: Управление, вычислительная техника и информатика. – 2021. – № 3. – С. 143-156. – https://doi.org/10.24143/2072-9502-2021-3-143-156.
7. Рогов И. Е., Адоньев А. А., Старичкова Ю. В. Опыт разработки, тенденции развития и внедрения информационных систем поддержки основного образовательного процесса / И. Е. Рогов, А. А. Адоньев, Ю. В. Старичкова // Современные информационные технологии и ИТ-образование. – 2017. – Т. 13, № 4. – С. 82-90. – https://doi.org/10.25559/SITITO.2017.4.628.
8. Мазорчук М. С., Тыжненко Ю. В. Автоматизированная информационная система "Деканат" // Объектные системы. – 2011. – № 3. – С. 52-58.