



**Уровень представления** обеспечивает визуализацию данных и знаний с учётом производственных норм и инструменты взаимодействия другими уровнями. При этом представление содержания репозитория и функций платформы должно осуществляться в соответствии с ролями пользователей. У платформы может быть большое число разных ролей в соответствии с должностными инструкциями работников предприятия.

С точки зрения функционирования платформы важно выделить два типа пользователя – конечного, который получает доступ к функциям платформы в соответствии с бизнес-процессом ТПП, и когнитолог/аналитик, пользователь, который отвечает за расширяемость репозитория платформы и контролирует процессы пополнения данных, знаний и текстов.

Макет гибкой платформы реализован на языках Python, JavaScript, HTML, использует машину вывода Pellet, взаимодействует с базой данных PostgreSQL и онтологией, хранящей информацию о предметной области поддержки технологического процесса. Средство разработки онтологии – Protege. Языки запросов к базе данных – SQL, к онтологии – SPARQL.

На языке Python реализуется серверная часть разработанной платформы и выполняются следующие функции: взаимодействие с базой данных; подключение к онтологии, созданной на языке OWL и сохранённой в формате RDF; генерацию и отображение HTML-страниц; создание и управление процессами, задачами, комплектами документов.

Для работы с онтологией используется дополнительная библиотека Owlready2, которая позволяет: импортировать онтологии в форматах NTriples, RDF/XML, OWL/XML; экспортировать онтологии в форматах NTriples, RDF/XML; управлять классами, экземплярами, свойствами онтологии как объектами Python; выполнять SPARQL запросы; запускать машины вывода Pellet, HermiT; сохранять полученные результаты в базе данных PostgreSQL.

Онтологическое моделирование на основе разработанной платформы осуществляется при решении ряда таких важных задач, как проверка комплектности технологической документации, формирование вида нового комплекта ТД, выбор оптимального комплекта ТД из множества релевантных прецедентных комплектов ТД, выявление проблемных ситуаций на основе выбранного релевантного комплекта ТД, повышение качества выпускаемых изделий на основе создания оптимальных маршрутных карт, поддержка инструментально-технологического сопровождения процессов проектирования шаблонной оснастки с использованием средств онтологической поддержки.

Использование онтологического моделирования в перечисленных задачах позволяет повысить качество ТПП.

А.А. Смагин, С.Н. Ларин, Р.Г. Бильданов, А.А. Булаев

## ПРОЕКТИРОВАНИЕ СРЕДСТВА ПРОВЕРКИ ВЫПОЛНЕНИЯ ЭТАПОВ ПРОИЗВОДСТВА РАДИОФАРМПРЕПАРАТОВ



(Ульяновский государственный университет)

Технологический процесс производства радиофармпрепаратов (РФП) включает в себя последовательно выполняемые операции, требующие проверки на корректное выполнение по нескольким показателям. Отслеживание вводимых показателей, а также возможность перехода от одной операции к другой контролируются с помощью средства проверки выполнения этапов производства РФП. Такое средство (Рис. 4) включает в себя следующие компоненты:

- *Оператора* (лаборанта), вводящего новые показатели по стадиям и операциям производства РФП;
- *онтологию*, описывающую процесс разработки РФП;
- *базу прецедентов*, содержащую данные о введенных показателях и результатах проверки;
- *имитатор* с графическим интерфейсом, обеспечивающий ввод новых показателей оператором (лаборантом) и выполняющий проверку введенных показателей.

Взаимодействие компонентов с имитатором осуществляется с помощью языков запросов к данным:

- *онтология и имитатор* – язык запросов SPARQL, работающий с данными, представленными по модели RDF;
- *база прецедентов и имитатор* – язык запросов SQL для взаимодействия с реляционными базами данных.

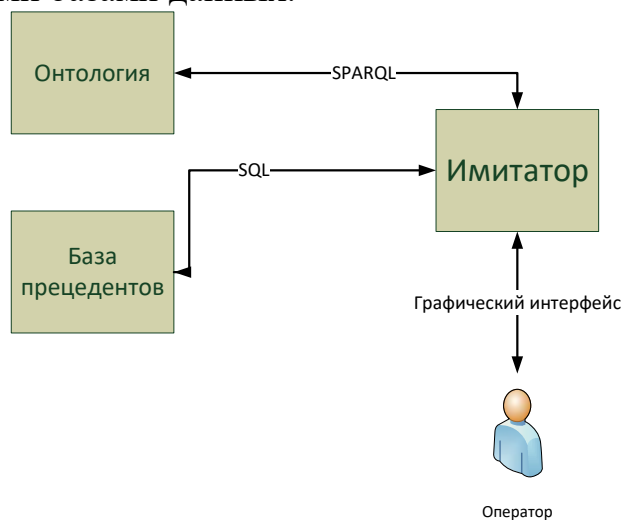


Рис. 4. Схема работы средства проверки выполнения этапов производства РФП

Имитатор (Рис. 5) выполняет следующие функции и состоит из модулей, осуществляющих эти функции:

- ввод оператором и проверка на корректность новых показателей;
- подключение к онтологии и обмен данными с ней по языку SPARQL;
- подключение к базе данных прецедентов и обмен данными с ней по языку SQL;



- формирование проверочных матриц на основе классов, объектов и их атрибутов, представленных в онтологии;
- проверка введенных оператором показателей и запись результатов в базу данных прецедентов.

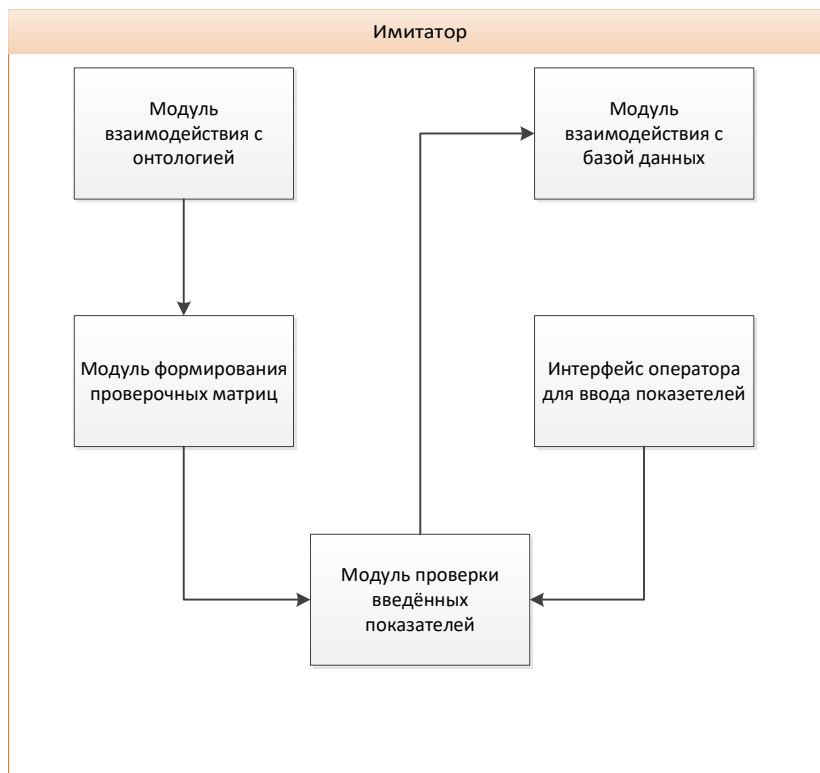


Рис. 5. Структурная схема имитатора

Ниже представлен алгоритм работы имитатора (Рис. 6), который включает в себя следующие шаги:

- подключение к онтологии и базе данных прецедентов;
- формирование проверочных матриц на основе данных из онтологии;
- ввод показателей с помощью интерфейса оператора (ввод показателей осуществляется путем установления «галочки» у тех операций производства, которые несут булевый характер (выполнено/не выполнено) и ввода численных значений у других показателей. В случае, если какое-то значение не было введено, то интерфейс «подсвечивает» красным цветом этот показатель и предлагает ввести его значение);
- проверка выполнения этапов производства путем перемножения вектора-строки введенных показателей на соответствующую проверочную матрицу;
- вывод результата проверки на экран и запись в базу данных прецедентов.

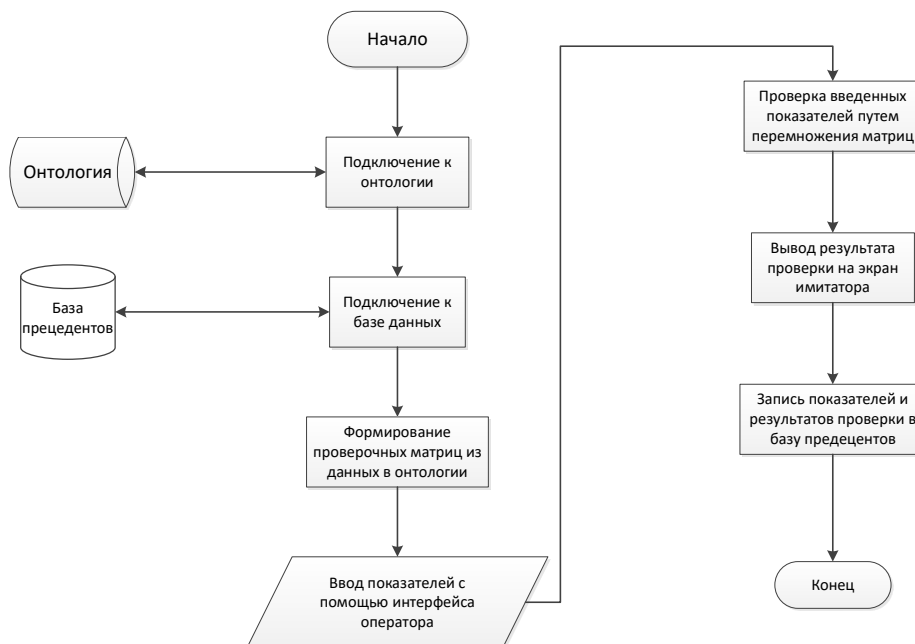


Рис. 6. Алгоритм работы имитатора

База данных прецедентов содержит 5 таблиц:

- операторы (пользователи), имеющие возможность вносить показатели;
- полный список операций производства РФП с привязкой по стадиям;
- истории выпусков РФП;
- вводимые ранее показатели по операциям производства РФП;
- выпуски РФП с описанием результата проверки, датой проверки и введенными при этой проверке показателями.

База данных прецедентов позволяет предсказывать возможность корректного производства РФП на основе частично введенных показателей и предыдущих проверок.

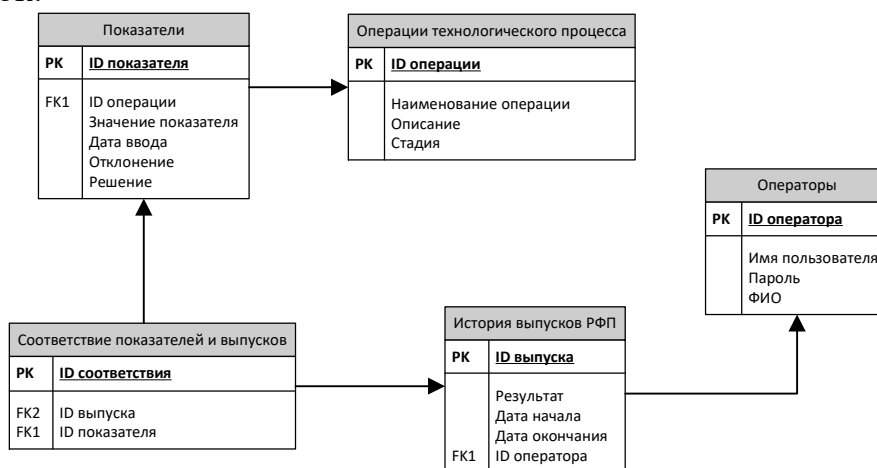


Рис. 7. Логическая модель базы данных прецедентов

Программная реализация базы данных прецедентов выполнена в СУБД PostgreSQL.

Онтология описания процесса разработки РФП содержит дерево стадий производства, операции, входящие в эти стадии, и показатели для каждой операции с информацией о допустимых значениях.



На основе данных, хранящихся в онтологии, строится графический интерфейс оператора в виде дерева стадий производства РФП и их операций. Для каждой операции отображаются элементы ввода по показателям (чекбокс, если показатель имеет булевый тип (да/нет), либо текстовое поле, если для показателя необходимо ввести числовое значение).

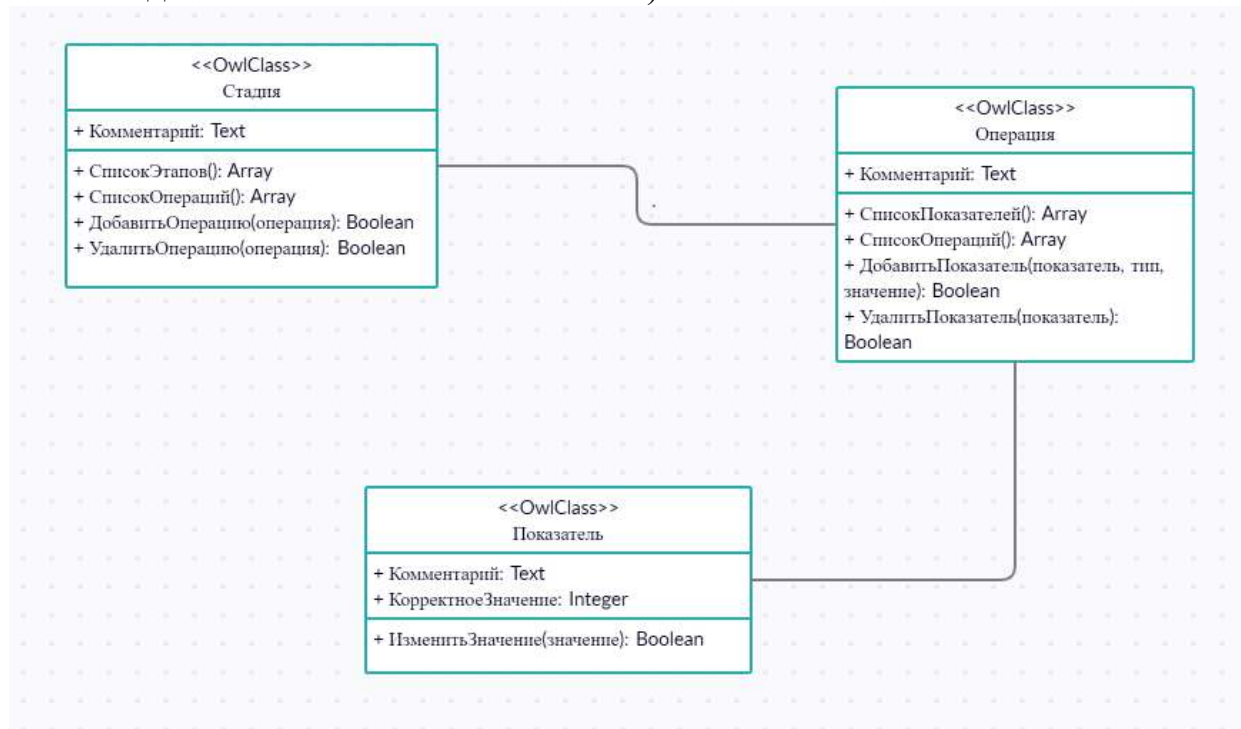


Рис. 8. Диаграмма классов онтологии описания процесса разработки РФП

Использование предложенного средства проверки выполнения этапов технологического процесса позволяет повысить качество производства РФП.

А.А. Сытник, Т.Э. Шульга

## МАТЕМАТИЧЕСКИЕ МЕТОДЫ МОДЕЛИРОВАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И АНАЛИЗА ДАННЫХ

(Саратовский государственный технический университет  
имени Гагарина Ю.А.)

Одной из актуальных проблем современной прикладной информатики является проблема моделирования и анализа больших объемов данных, представленных в электронном виде. В рамках этой проблемы особое место занимает задача анализа данных с целью поддержки принятия решений. Традиционная математическая статистика, долгое время претендовавшая на роль основного инструмента анализа данных, сегодня в основном используется для проверки заранее сформулированных гипотез и для “грубого” разведочного анализа, составляющего основу оперативной аналитической обработки данных (OLAP). Для выявления неизвестных, нетривиальных, практически полезных и доступных для интерпретации знаний, необходимых для принятия решений в различ-