

МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное учреждение высшего образования
«КРАСНОЯРСКИЙ ГОСУДАРСТВЕННЫЙ ПЕДАГОГИЧЕСКИЙ
УНИВЕРСИТЕТ им. В.П. АСТАФЬЕВА»
(КГПУ им. В.П. Астафьева)

Институт математики, физики и информатики
Выпускающая кафедра: информатики и информационных технологий в
образовании

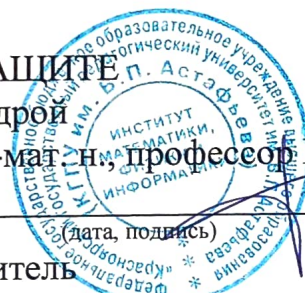
Голубков Константин Алексеевич

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

**Агент для тестирования уровня сформированности комплексных
понятий школьного курса информатики**

Направление подготовки:
44.03.05 Педагогическое образование (с двумя профилями подготовки)
Направленность (профиль) образовательной программы:
Математика и Информатика

ДОПУСКАЮ К ЗАЩИТЕ
Заведующий кафедрой
д-р пед. н., к. физ.-мат. н., профессор Н.И. Пак



(дата, подпись)

Научный руководитель
к. физ.-мат. н., доцент Д.В. Романов

Романов

(дата, подпись)

Дата защиты

22.06.2026

Обучающийся
К. А. Голубков

Голубков

(дата, подпись)

Оценка *отлично*

прописью

Красноярск 2026

Оглавление

Введение.....	3
Глава 1. Теоретические основы применения ИИ-агентов для анализа сформированности комплексных понятий школьного курса информатики ..	12
1.1 Методология применения ИИ-агентов в процессах информационной обработки ..	12
1.2 Автоматизация деятельности учителя при анализе степени сформированности комплексных понятий школьного курса информатики: теоретико-методологические и практические аспекты ..	28
1.3 Обоснование архитектуры и функциональных требований к ИИ-агенту в контексте профессиональной деятельности педагога: теоретико-методологический и технологический аспекты.....	43
Вывод по первой главе.....	58
Глава 2. Проектирование и апробация программного решения для автоматизации диагностики знаний.....	61
2.1. Разработка системы команд и навыков ИИ-агента для диагностики сформированности понятий школьного курса информатики ..	61
2.2. Установка и запуск ИИ-агента ..	76
Заключение ..	87
Список использованных источников ..	90

Введение

Профессиональная деятельность педагога относится к числу интеллектуально и когнитивно насыщенных видов труда, поскольку на этапах планирования, реализации и контроля учебного процесса преподаватель вынужден одновременно учитывать факторы, принадлежащие к принципиально разным системам:

- административно-регулятивной, закреплённой в законодательных и нормативно-правовых документах; профильного министерства,
- выступающего в роли заказчика и разработчика непрерывно корректируемых требований к образовательным результатам;
- непосредственно учебной группы (класса обучающихся) со всеми её психолого-педагогическими характеристиками – состоянием, мотивацией и уровнем знаний как отдельного учащегося, так и класса в целом, а также взаимодействием с родителями и коллегами.

Содержание образовательного результата в значительной степени определяется требованиями государства и экономики, которые изменяются с нарастающей динамикой под воздействием внутренних и внешних факторов. Традиционные инструменты управления этим процессом (повышение квалификации и корректировка учебных программ) объективно не успевают за скоростью трансформаций: технологии генеративного искусственного интеллекта за период с декабря 2022 г. по июнь 2026 г. эволюционировали от статуса неизвестного стартап-проекта до уровня нескольких конкурирующих корпораций с объёмами инвестиций, исчисляемыми десятками триллионов

долларов, чьё развитие напрямую координируется правительствами США и Китая. За тот же временной интервал студенты, поступившие в педагогические вузы, не успели завершить обучение, а содержание их образовательных программ не успело актуализироваться. Аналогичная картина наблюдается и в других секторах экономики развитых стран: инструменты искусственного интеллекта были интегрированы в действующие технологические цепочки без предварительной оценки последствий и системного контроля над процессом внедрения.

Наиболее острое противоречие заключается в том, что технологии искусственного интеллекта либо вытесняют специалистов умственного труда, либо предоставляют им принципиально новые и высокоэффективные средства работы. Учащийся, получив задание по математическому анализу, фотографирует условие и в течение нескольких секунд получает полный разбор решения с методическими комментариями, а при необходимости – голосовое объяснение сложного фрагмента, иллюстративные примеры и тренажёр для отработки выявленного пробела, причём всё это доступно с обычного мобильного телефона. Расхождение между потенциальными возможностями новых технологий и практически полным отсутствием в образовательных учреждениях видения, разработанных методик и сложившейся культуры их использования указывает на актуальность настоящего исследования.

Данное противоречие проявляется с особой остротой в профессиональной деятельности учителя информатики, в частности при диагностике уровня сформированности комплексных понятий школьного курса, таких как «алгоритм», «система счисления», «информационная модель», «структура данных» и «рекурсия». Эти категории представляют собой не просто набор терминов, а комплексные образования, требующие от обучающихся развитого абстрактно-логического мышления: их полноценное усвоение предполагает не репродуктивное запоминание определений, а глубокое понимание внутренних логических связей, операционализацию знаний и перенос усвоенных когнитивных структур в новые, нестандартные контексты.

Традиционные методы педагогической диагностики, ориентированные на проверку фактологических знаний и способности к их воспроизведению, упускают возможность понять, как на самом деле формируются и развиваются понятия у учащихся. Существующие контрольно-измерительные материалы не позволяют в полном объёме сопровождать индивидуальные образовательные траектории, своевременно диагностировать скрытые пробелы в ментальных схемах обучающихся и объективно оценивать степень усвоения абстрактных информационных моделей. Учитель информатики, сталкиваясь с постоянным ростом учебного материала, разнообразием форматов его представления и жёсткими хронометражными рамками, испытывает значительную когнитивную и эмоциональную нагрузку, что ведёт

к снижению эффективности формирующего оценивания и затрудняет своевременную, адресную педагогическую коррекцию.

Теоретические разработки современной педагогики, описывающие внутреннюю структуру понятия и способы её анализа, остаются в силу своей абстрактности и трудоёмкости применения недоступными для рядового учителя, ограниченного во времени и не обладающего навыками самостоятельной автоматизации диагностических процедур, притом что в педагогической и психологической науке уже сформирован обширный корпус теоретических описаний и диагностических процедур, исследующих механизмы усвоения сложных концептов. Деятельностный подход А. Н. Леонтьева, постулирующий неразрывное единство сознания и деятельности, и культурно-историческая концепция Л. С. Выготского подчёркивают фундаментальную роль опосредствующих инструментов в расширении зоны ближайшего развития и интериоризации знаний.

В развитие этой мысли появление больших языковых моделей (Large Language Model, далее LLM) и на их основе автономных ИИ-агентов создало принципиально новый класс таких опосредствующих инструментов. Современные LLM способны интерпретировать сложные научные тексты, извлекать из них логически корректные выводы и направлять их на достижение заранее заданной педагогической цели. Тем самым теоретические положения Выготского и Леонтьева получают технологическую реализацию: ИИ-агент выступает не как внешнее средство автоматизации рутины, а как

когнитивный посредник, способный перерабатывать классические педагогические и психологические источники, интегрировать их с актуальными данными (ответами обучающихся, рабочими программами, структурно-ментальными схемами) и предлагать конкретные варианты решения педагогических задач от диагностики уровня сформированности понятия до генерации дифференцированных корректирующих заданий.

В данном парадигмальном поле автономные ИИ-агенты рассматриваются как полноценные когнитивные посредники, способные преодолеть указанный теоретико-практический разрыв. Они обеспечивают переход от статичной суммативной диагностики к динамическому, многоуровневому мониторингу сформированности комплексных понятий, опирающемуся на ментальный подход Н. И. Пака и теорию построения структурно-ментальных схем, что позволяет визуализировать процесс мышления ученика.

Гипотеза работы: ряд актуальных теоретических наработок современных педагогов, недоступных для рядового учителя в силу абстрактного характера и высокой теоретической сложности с одной стороны, и нехватку сил, времени и квалификации с другой, теперь можно внедрять в учебные процессы автоматизированно, с помощью технологий агентского программирования. Это снимает с учителя необходимые требования к научной стороне работы, и при этом даёт им инструменты, результаты работы которых жизненно необходимы для многогранной работы педагога в

постоянно меняющихся условиях (что ещё раз подчёркивает актуальность настоящего исследования).

Проблема: уровень сложности и абстрактности ряда современных работ по педагогике превышают возможности школьного учителя по их усвоению и адаптации (учитывая текущий уровень и характер нагрузок в условиях постоянных изменений)

Цель: переложить задачу по адаптации современных работ теоретической педагогики к учебному процессу школы на ИИ-агента, предоставив учителю готовые наукоёмкие результаты в форме, удобной для использования в учебном процессе.

Предмет: автоматизация диагностики уровня сформированности комплексных понятий школьного курса информатики у обучающегося.

Объект: процесс обучения информатике в школе.

Для достижения цели необходимо решить следующие **задачи:**

1. Исследовать методики применения агентов для решения задач обработки информации, создания документов и написания кода.
2. Выделить основные этапы работы учителя, требуемые ему артефакты и сформулировать требования к ИИ агенту.
3. Обосновать выбор архитектуры и функциональных возможностей ИИ-агента путём алгоритмизации деятельности учителя при диагностике знаний.

4. Разработать систему команд и навыков ИИ-агента для проведения диагностики сформированности понятий.
5. Реализовать программный прототип ИИ-агента и описать процесс его развертывания и настройки в локальной вычислительной среде.

Методологическую основу исследования составили фундаментальные труды Л. С. Выготского, А. Н. Леонтьева, А. В. Усовой и Н. И. Пака, заложившие базис понимания процессов усвоения научных понятий, а также современные исследования в области агентного искусственного интеллекта и педагогической информатики. В работе использован комплекс взаимодополняющих методов: теоретический анализ и синтез научно-педагогической литературы для выявления ключевых закономерностей; сравнительно-сопоставительный метод для оценки эффективности различных диагностических подходов; алгоритмическое моделирование педагогических процессов для формализации деятельности учителя.

Научная новизна настоящего исследования заключается в разработке целостной методологии интеграции автономных ИИ-агентов в диагностическую деятельность педагога, базирующейся на глубоком синтезе алгоритмизации профессиональных действий учителя и ментальной таксономии учебных понятий, что позволяет визуализировать ранее интуитивные процессы оценивания. На основе данной методологии был

создан и эмпирически верифицирован комплекс специализированных промпт-навыков агента, обеспечивающих не только построение динамических семантических графов знаний, но и автоматизированную генерацию персонализированных корректирующих заданий. Наряду с этим, в работе предложено педагогически ориентированная архитектура ИИ-системы, в которой императивно закреплены принципы интерпретируемости генерируемых решений, сохранения человеческого контроля в контуре принятия педагогических решений и безусловного приоритета конфиденциальности персональных данных обучающихся, что принципиально отличает разработанный подход от существующих коммерческих аналогов.

Практическая значимость определяется созданием воспроизводимого инструмента с открытым исходным кодом (open-source), готового к тиражированию и внедрению в практику педагогических вузов и общеобразовательных организаций. Разработанный ИИ-агент позволяет существенно снизить рутинную административную и диагностическую нагрузку учителя, повысить объективность и точность формирующего оценивания, а также поддержать реальную индивидуализацию обучения, строго соотнесенную с актуальной зоной ближайшего развития конкретного обучающегося, минимизируя риски алгоритмической предвзятости.

Работа включает введение, две главы, заключение, список использованных источников и приложения. В первой главе изложены

теоретико-методологические основы исследования и проведен глубокий анализ состояния проблемы формирования комплексных понятий в современной школе. Во второй главе представлены результаты проектирования, программной реализации и развертывания, разработанного ИИ-агента.

Глава 1. Теоретические основы применения ИИ-агентов для анализа сформированности комплексных понятий школьного курса информатики

1.1 Методология применения ИИ-агентов в процессах информационной обработки

В эпоху стремительной цифровой трансформации образования, активного внедрения так называемых сквозных технологий (включая большие данные, нейросетевые архитектуры и распределённые реестры) и непрерывного совершенствования интеллектуальных систем искусственный интеллект (ИИ) уверенно занимает позицию одного из ведущих инструментов для автоматизированной работы с разнородными информационными массивами. Современный этап научно-технического прогресса характеризуется не только лавинообразным ростом объёмов данных, но и существенным усложнением их внутренней структуры, многообразием форматов представления, а также острой потребностью в оперативном выявлении смысловых и ассоциативных связей в условиях высокой неопределённости и постоянной изменчивости педагогического контекста. В этой связи особую актуальность приобретают специализированные ИИ-агенты, которые выступают не просто в роли вычислительных блоков, а как полноценные когнитивные посредники, способные кардинально трансформировать традиционные подходы к информационному обмену, учебной коммуникации и научно-методическому анализу.

В современной научно-исследовательской литературе под ИИ-агентами, как правило, понимаются автономно действующие программно-

алгоритмические комплексы, обладающие способностью проводить многофакторный анализ поступающей информации, принимать обоснованные целевые решения, выстраивать осмысленный диалог с конечными пользователями (в число которых входят педагоги, методисты, исследователи и администраторы образовательных систем), а также гибко адаптироваться к изменяющимся условиям внешней информационной среды [35]. В отличие от детерминированных экспертных систем, функционирующих на основе жёстко заданных правил, ИИ-агенты демонстрируют черты автономного поведения, опираясь на комплекс методов машинного обучения, обработки естественного языка (NLP), символического логического вывода и взаимодействия с внешними цифровыми ресурсами и сервисами. В образовательной сфере такие системы могут выступать не только как инструменты анализа учебных данных, но и как активные участники педагогического процесса: они способствуют персонализации обучения, автоматизации формирующего оценивания, поддержке индивидуальных образовательных траекторий и повышению эффективности научно-методической работы преподавателей [16]. При этом, как обоснованно отмечается в исследованиях, посвящённых типологии интеллектуальных агентов, их внедрение связано не только с расширением функциональных возможностей образовательных платформ, но и с серьёзными методологическими вызовами, требующими пересмотра традиционной роли педагога, форматов учебного взаимодействия и подходов к проектированию образовательного контента [19].

Практическое применение и системная интеграция ИИ-агентов в действующие образовательные и технологические экосистемы позволяют заметно повысить общую результативность обработки информации. Такой положительный эффект достигается за счёт глубокой автоматизации рутинных, интеллектуально насыщенных и трудоёмких операций (например, проверки домашних заданий, подбора индивидуальных упражнений, составления аналитических справок), а также значительного ускорения выработки и принятия управленческих и методических решений. Успешная апробация данного подхода подтверждена в таких разнородных областях, как современное образование [6, 14], прикладная медицина [24] и комплексная информационная безопасность [29]. Педагогическая работа относится к высокоинтеллектуальным видам деятельности: на этапах планирования, проведения и контроля занятий преподаватель вынужден согласовывать требования из принципиально разных сфер – законодательно-нормативной базы, постоянно обновляемых министерских стандартов, а также психолого-педагогических особенностей конкретной учебной группы (уровень знаний, мотивация, состояние учащихся, их индивидуальные различия) и взаимодействия с родителями и коллегами.

Образовательные результаты всё сильнее зависят от запросов государства и экономики, которые меняются с нарастающей скоростью под влиянием внутренних и внешних импульсов. Традиционные механизмы реакции – повышение квалификации и корректировка программ – объективно

не успевают за этим темпом. Пример – генеративный ИИ: с декабря 2022 по июнь 2026 года он превратился из малоизвестного проекта в сферу конкуренции корпораций с триллионными инвестициями, причём развитие координируется на уровне правительств США и Китая. За тот же период студенты педагогических вузов ещё не завершили обучение, а содержание их учебных планов не обновилось. Аналогичная ситуация сложилась и в других отраслях развитых стран – ИИ-инструменты внедрялись в технологические цепочки без предварительной оценки рисков и системного контроля.

Главная дилемма: ИИ либо вытесняет работников умственного труда, либо даёт им принципиально новые, сверхэффективные средства. Школьник, получив задание по математическому анализу, может сфотографировать условие и через несколько секунд получить полный разбор решения с методическими пояснениями, голосовыми комментариями, примерами и тренажёром для отработки пробелов – всё это доступно через обычный смартфон. Однако разрыв между этими технологическими возможностями и почти полным отсутствием в образовательных учреждениях концептуальных подходов, отработанных методик и устоявшейся культуры использования ИИ делает данное исследование крайне актуальным.

Особенно остро это противоречие проявляется в деятельности учителя информатики при диагностике сложных понятий школьного курса – таких как «алгоритм», «система счисления», «информационная модель», «структура данных», «рекурсия». Это не просто термины, а целостные конструкты,

требующие от учащихся развитого абстрактно-логического мышления: усвоение предполагает не заучивание определений, а понимание внутренних логических связей, умение оперировать знаниями и переносить освоенные ментальные схемы в новые нестандартные контексты.

Существующие диагностические средства ориентированы в основном на проверку фактологических знаний и их воспроизведения, поэтому они не дают представления о реальном процессе формирования понятий. Они не позволяют в полной мере сопровождать индивидуальные образовательные траектории, своевременно выявлять скрытые дефекты в ментальных моделях учащихся и объективно оценивать глубину усвоения абстрактных информационных конструкций. Учитель информатики, сталкиваясь с постоянно растущим объёмом материала, разнообразием форматов и жёстким хронометражем, испытывает значительную когнитивную и эмоциональную перегрузку, что снижает эффективность формирующего оценивания и осложняет своевременную адресную коррекцию.

В педагогической и психологической науке накоплен обширный теоретический аппарат, описывающий структуру понятий и методы их анализа, однако эти разработки остаются для рядового учителя малодоступными из-за абстрактности, трудоёмкости и недостатка времени, а также отсутствия навыков алгоритмизации диагностических процедур. При этом фундаментальные подходы – деятельностная теория А.Н. Леонтьева и культурно-историческая концепция Л.С. Выготского – подчёркивают

решающую роль опосредствующих инструментов в расширении зоны ближайшего развития и интериоризации знаний.

В развитие этих идей появление больших языковых моделей (LLM) и созданных на их основе автономных ИИ-агентов формирует принципиально новый класс опосредствующих средств. Современные LLM способны интерпретировать сложные научные тексты, извлекать логически корректные выводы и направлять их на решение педагогических задач. Тем самым теоретические положения Выготского и Леонтьева получают технологическую реализацию: ИИ-агент выступает не просто как средство автоматизации, а как когнитивный посредник, перерабатывающий классические источники, актуальные данные (ответы учащихся, рабочие программы, структурно-ментальные схемы) и предлагающий конкретные варианты – от диагностики уровня усвоения понятий до генерации дифференцированных корректирующих заданий.

В данной парадигме автономные ИИ-агенты рассматриваются как полноценные когнитивные медиаторы, способные преодолеть описанный теоретико-практический разрыв. Они обеспечивают переход от статичной суммативной оценки к динамическому многоуровневому мониторингу сформированности комплексных понятий, опирающемуся на ментальный подход Н.И. Пака и теорию структурно-ментальных схем, что позволяет наглядно визуализировать мыслительные процессы ученика.

Такое перераспределение ролей особенно актуально в условиях перехода от знаниевой парадигмы к компетентностной, где на первый план выходят не столько объём усвоенной информации, сколько способность действовать в нестандартных профессиональных ситуациях. Следовательно, методология использования ИИ-агентов должна включать не только технические инструкции, но и педагогические сценарии совместной деятельности, а также критерии оценки эффективности такого партнёрства с точки зрения развития профессионального мышления будущего учителя.

В глобальной гонке ИИ картина складывается неоднозначная. Коммерческий рывок OpenAI и общее лидерство США по передовым моделям пока не вызывают сомнений – они заметно обгоняют китайские разработки, хотя КНР, что немаловажно, сокращает разрыв очень активными темпами. Так что два главных игрока на этом поле США и Китай. А что Европа? Научная школа там крепкая, фундаментальная, но вот с вычислительными ресурсами и объёмами реальных данных европейцы серьёзно проигрывают.

США, по сути, ушли от России в развитии ИИ на целую технологическую эпоху. И они же первыми среди крупных обществ прочувствовали на себе весь «удар» больших языковых моделей, в первую очередь из-за массового распространения ChatGPT в англоязычной среде, где языковой барьер просто не мешал внедрению. А раз так, мы можем извлечь ценный урок из тех социальных и экономических последствий, с которыми американцы уже столкнулись. Примечательно, что европейский опыт для нас

едва ли не важнее. Почему? Потому что Европа первой сместила фокус не на скорость разработок, а на анализ последствий и регуляторные механизмы. Они скрупулёзно выстроили юридическую базу, с упором на защиту прав и этические рамки. Это тот случай, когда к их наработкам стоит присмотреться очень внимательно.

Обращаясь к нормативно-методологической базе, регулирующей описанный технологический стек, следует указать на документ NIST AI 100-1 Artificial Intelligence Risk Management Framework (AI RMF 1.0). [35]. Под искусственным интеллектом понимают специализированную машинную систему, которая действует с разной степенью автономности. Её задача: на основе чётко заданных целей и входных параметров выдавать точные прогнозы, обоснованные рекомендации или готовые практические решения.

Получаемые на выходе артефакты ИИ-систем призваны оказывать прямое или косвенное влияние на свойства как реальной (физической), так и виртуальной (цифровой) среды. Данный нормативный подход особо подчёркивает фактор автономности работы интеллектуальных систем, а также их способность эффективно заменять или дополнять аналитические и когнитивные функции, которые ранее были исключительной прерогативой человека. В контексте обработки информации ИИ-агенты выступают в роли высокотехнологичных программных компонентов, реализующих алгоритмы интеллектуального анализа данных (Data Mining), извлечение информации из текстов на естественном языке, автоматизированный поиск скрытых

закономерностей в неструктурированных массивах данных, прогностическое формирование содержательных выводов и диагностических отчётов на основе технологий Big Data [4]. В развитие этой мысли можно отметить, что в образовании такие возможности позволяют перейти от статичной диагностики знаний к динамическому мониторингу процесса формирования понятийных структур и движения по индивидуальным образовательным траекториям, что полностью соответствует принципам культурно-исторической психологии Л.С. Выготского [5], рассматривавшего развитие как опосредованный процесс, активизируемый в зоне ближайшего развития с помощью внешних средств и социальных взаимодействий [18].

Кроме того, важным направлением развития методологии ИИ-агентов становится обеспечение их объяснимости (explainability) и интерпретируемости для педагогов и обучающихся, не обладающих глубокими знаниями в области машинного обучения. Многие современные модели, особенно глубокие нейронные сети, работают по принципу «чёрного ящика», что порождает недоверие со стороны профессионального сообщества и ограничивает их внедрение в образовательную практику. Для преодоления этого барьера необходима разработка специализированных интерфейсов, которые визуализируют логику принятия решений агентом: например, показывают, на основании каких фрагментов текста был сделан тот или иной вывод, какие альтернативные варианты рассматривались и с какой степенью уверенности. В педагогическом контексте это особенно значимо для

формирующего оценивания, где важно не просто получить итоговую оценку, но и понять процесс её формирования, чтобы скорректировать учебные действия. Поэтому в методологию ИИ-агентов следует включать требования к проектированию «прозрачных» архитектур, а также к обучению пользователей навыкам критической интерпретации агентных рекомендаций, что снижает риски некритичного принятия ошибочных или предвзятых решений.

Методология практического использования и встраивания ИИ-агентов в контур обработки информации должна в идеале представлять собой развёрнутую совокупность научных принципов, алгоритмических методов и организационно-управленческих механизмов. Внедрение этой методологии должно быть нацелено на обеспечение стратегических ориентиров: надёжности, прозрачности алгоритмов, конфиденциальности, информационной безопасности и управляемости рисков при взаимодействии человека с искусственным интеллектом. Системное проектирование педагогических, аналитических и информационных процессов с привлечением таких агентных инструментов должно опираться на строго гуманистический подход, что предполагает приоритет человеческого фактора, этическую ответственность и ориентацию на развитие субъектности обучающихся [17]. Это требует обязательной защиты персональных данных, соблюдения возрастных ограничений, построения человекоцентричной архитектуры. На методологическом уровне это означает, что ИИ-агент не

заменяет педагога, а выступает в роли когнитивного соавтора, чья деятельность должна находиться под постоянным профессиональным контролем и методологической рефлексией преподавателя. Более того, в условиях педагогического вуза важно формировать у будущих учителей понимание границ применимости ИИ-агентов, их возможных ошибок («галлюцинаций») и методологии критической оценки получаемых результатов.

Наконец, нельзя обойти стороной временные и ресурсные издержки, связанные с интеграцией ИИ-агентов в учебный процесс, – аспект, который в теоретических исследованиях нередко остаётся в тени, однако на практике играет определяющую роль. Внедрение агентных систем сопряжено не только с приобретением или созданием софта, но и с развёртыванием вспомогательной инфраструктуры: вычислительных мощностей, защищённых каналов передачи данных, надёжных хранилищ, – а также требует существенных затрат времени на повышение квалификации сотрудников, переработку учебных программ и адаптацию методических пособий. Поэтому стратегия применения ИИ-агентов обязана включать финансово-экономическое обоснование и поэтапные планы внедрения, соотнесённые с реальными возможностями конкретного вуза. Для педагогических университетов, как правило, характерны скромный бюджет и высокая преподавательская занятость, а значит, предпочтение стоит отдавать открытым решениям и модульным архитектурам, которые позволяют

расширять функционал постепенно, избегая единовременных крупных расходов. Подобный прагматичный подход укрепляет жизнеспособность нововведений и уменьшает вероятность отказа от ИИ-агентов после первых неудач.

Опора на открытый программный стек (например, модели семейства Qwen под управлением Ollama на Python) выводит обработку информации на качественно иную ступень инструментальной автономии. В отличие от базовых больших языковых моделей (LLM), работающих по принципу «мозг в банке», ИИ-агенты реализуют инструментальную парадигму, предоставляя модели доступ к файловой системе, внешним библиотекам, интерфейсам прикладного программирования и специализированным протоколам взаимодействия (включая MCP-серверы) [34]. Благодаря такой архитектуре агент не ограничивается выдачей рекомендаций, а самостоятельно производит прикладные технические действия: читает документы в разных форматах (PDF, DOCX, TXT), проверяет наличие нужных библиотек в программной среде, создаёт файлы с аналитическими выкладками и визуализирует полученные результаты (к примеру, строит семантические графы понятий с сохранением в PNG). Подобная технологическая организация гарантирует институциональную независимость данных, сокращает зависимость от внешних облачных сервисов и обеспечивает воспроизводимость исследовательской среды, что становится критически важным для

педагогических вузов, имеющих дело с конфиденциальными сведениями – личными данными учащихся и материалами тестирований.

Ключевым методологическим подходом здесь выступает рассмотрение агентного приложения в роли интеллектуального диспетчера, координирующего обработку пользовательских запросов. Для обеспечения надёжного функционирования задействуются такие паттерны, как ReAct (рассуждение и действие), цепочка логических умозаключений (Chain-of-Thought), генерация с дополненным извлечением (RAG) [32, 39, 41], а также архитектуры с множеством взаимодействующих агентов. Указанные техники дают возможность расчленять сложные педагогические и научно-исследовательские вопросы на отдельные задачи, привлекать внешние ресурсы (базы данных по предметным областям, поисковые сервисы, инструменты верификации данных) и достигать высокого качества решений при минимальном участии человека. В образовательной сфере подобные шаблоны представляют особую значимость для оценки учебных достижений и автоматизации диагностических процедур: RAG даёт опору на достоверные методические материалы и регламентирующие документы, CoT воспроизводит поэтапную логику педагогического размышления, сходную с реальным ходом мыслей преподавателя, а ReAct организует циклы «анализ → действие → обратная связь», что перекликается с логикой формирующего оценивания. В то же время интеграция многоагентных архитектур (с агентами-супервизорами, методологами и аналитиками) усиливает координацию и

методологическую поддержку процессов, что соответствует современным рекомендациям по управлению научно-исследовательской деятельностью и принципу распределённого когнитивного труда в профессиональных педагогических сообществах [16, 19].

Методология использования ИИ-агентов предполагает непрерывный мониторинг, валидацию и оценку эффективности системы на всех этапах её жизненного цикла - от проектирования до промышленной эксплуатации. Согласно стандартам ISO/IEC 22989:2022 и ISO/IEC TS 5723:2022, измерение характеристик ИИ (надёжность, предвзятость, точность, устойчивость к смещению распределений данных) должно носить системный и регулярный характер. Это позволяет своевременно выявлять артефакты генераций (так называемые «галлюцинации»), адаптировать систему под меняющиеся образовательные стандарты и требования ФГОС, а также поддерживать высокое качество аналитики в условиях постоянного роста объема учебных и методических данных. Особое внимание заслуживает аспект кибербезопасности: в агентных системах риски, связанные с состязательными атаками, подменой контекста и внедрением вредоносных промптов, существенно выше, чем в классических информационных системах, поэтому необходима имплементация специальных защитных мер – фильтрации входящих запросов, проверки корректности логических цепочек, разграничения доступа к инструментарию и детального логирования всех этапов взаимодействия [13]. Применительно к педагогической сфере это

означает, что требуется создание внутривузовских нормативных документов по использованию ИИ, регулярное тестирование учебных сценариев с помощью методики «красной команды» (моделирование атак и нестандартных ситуаций), а также планомерное развитие у преподавателей умений критически осмысливать и оценивать результаты, выдаваемые агентами.

Наряду с этим, внедрение ИИ-агентов в образование сопряжено с рядом системных вызовов: преодоление инфраструктурных и кадровых барьеров цифровизации (нехватка вычислительных ресурсов, низкая ИИ-грамотность части сотрудников), обеспечение этической безопасности и защита от алгоритмической дискриминации, повышение ИИ-грамотности педагогов через программы повышения квалификации, а также минимизация рисков алгоритмической предвзятости, связанной с нерепрезентативными учебными выборками. Перспективы дальнейшего развития связаны с созданием многоагентных экосистем, интеграцией с цифровыми двойниками образовательных процессов (виртуальными моделями классов, учебных групп и индивидуальных траекторий) и разработкой специализированных агентных модулей для поддержки методической работы преподавателей - например, автоматических конструкторов рабочих программ, интеллектуальных помощников по подбору дидактических материалов и систем анализа уязвимостей в учебных планах. Такой подход полностью соответствует стратегии цифровой трансформации российского образования, требованиям

ФГОС и способствует формированию у будущих педагогов компетенций 21 века, включая цифровую рефлексию, проектное мышление и способность к эффективному человеко-машинному кооперативному взаимодействию [18].

Резюмируя вышесказанное, методология автономных ИИ-агентов представляет собой высокотехнологичный комплексный подход к интеллектуальной обработке данных, основанный на принципах системности, безопасности, прозрачности и сквозного управления рисками на всех этапах - от сбора данных до интерпретации результатов. В практическом плане она трансформирует пассивное получение информационных ответов (например, в виде текстовых справок) в активное применение инструментами анализа, обратную связь через интерактивные отчёты, автоматизированное выполнение цепочек операций, существенно повышая качество структурирования знаний, точность педагогической диагностики, и снижая риски интеграции генеративного ИИ в профессиональную педагогическую и научно-исследовательскую деятельность [4, 16]. Устойчивое внедрение данной методологии потребует согласования технологических возможностей с педагогическими целями, непрерывного методологического сопровождения, включая повышение квалификации преподавателей, и сохранения приоритета развивающей функции образования над чисто инструментальной эффективностью.

1.2 Автоматизация деятельности учителя при анализе степени сформированности комплексных понятий школьного курса информатики: теоретико-методологические и практические аспекты

Разработка интеллектуальных агентов, ориентированных на поддержку педагогической деятельности, неизбежно ставит перед исследователем задачу формализации профессиональных действий учителя. Прежде чем передать часть аналитических и диагностических функций программной системе, необходимо детально описать эти функции в виде, пригодном для алгоритмической реализации. Именно поэтому алгоритмизация деятельности учителя рассматривается в настоящем исследовании не как самоцель, а как обязательный предварительный этап проектирования ИИ-агента, предназначенного для диагностики степени сформированности комплексных понятий школьного курса информатики.

Алгоритмизация педагогической деятельности позволяет представить процесс анализа сформированности понятий как упорядоченную последовательность операций с чёткими условиями перехода между этапами и разветвлениями в зависимости от актуального уровня подготовки учащихся. Такой подход обеспечивает двойной эффект: с одной стороны, он делает профессиональные действия учителя воспроизводимыми и доступными для программной реализации, с другой придаёт диагностическому процессу системный характер, исключая произвольные пропуски в оценке понятийного аппарата школьника.

Деятельность учителя информатики при работе с такими комплексными понятиями, как «алгоритм», «вычислительное мышление», «информационная

модель», «структура данных», «ментальная схема» и другие носит строго целенаправленный, поэтапный и непрерывно рефлексивный характер. Эти операции направлены: 1) первичная диагностика актуального уровня сформированности понятийного аппарата обучающегося; 2) последующее структурирование содержания понятия; 3) проектирование дидактических условий для его глубокого усвоения; 4) объективную оценку достигнутых результатов.

Каждый из этих этапов может быть представлен в виде алгоритма с разветвлениями. Например, на этапе диагностики учитель проверяет, есть ли у учащихся житейские представления о моделировании: если да, то опирается на них, если нет, то вводит понятие через конкретный пример (план класса, схема метро). Подобная алгоритмическая гибкость позволяет адаптировать обучение к реальному уровню класса, не снижая при этом системности педагогического воздействия. Именно подобная алгоритмизация становится основой для последующей передачи диагностических функций интеллектуальному агенту.

Проектирование таких алгоритмизированных сценариев учебной деятельности находит отражение в технологических картах уроков информатики, методологические основы разработки которых подробно рассмотрены в исследованиях И. В. Левченко, А. Р. Садыковой, Л. И. Карташовой и П. А. Меренковой [11]. Именно подобная алгоритмизация

становится основой для последующей передачи диагностических функций интеллектуальному агенту.

Важно отметить, что алгоритмизация деятельности учителя является не конечной целью, а необходимым условием её автоматизации. Как только педагогические действия по разбору комплексного понятия описаны в виде формализованных алгоритмов с явными условиями, критериями и переходами, они становятся доступны для воспроизведения интеллектуальным агентом. Это означает, что повторяющиеся, структурно однородные операции – построение понятийного паспорта, последовательное прохождение уровней таксономии, выбор типа диагностической задачи – могут быть переданы агентной системе, освобождая когнитивные ресурсы учителя для тех аспектов педагогической деятельности, которые требуют живого профессионального суждения и не поддаются полной формализации.

В следствии этого, автоматизация деятельности учителя при анализе комплексных понятий – это не замена педагога, а целенаправленное перераспределение интеллектуальной нагрузки между учителем и агентной системой.

Чтобы такое перераспределение нагрузки между учителем и агентом могло быть реализовано на практике, необходимо прежде всего точно определить, какой объект подлежит формализованному анализу в рамках разрабатываемой агентной системы. Центральным объектом диагностики в

данном контексте выступает «комплексное понятие» – особая единица научного знания, отличающаяся многоуровневой внутренней организацией. В современной педагогической методологии комплексное понятие трактуется не как статичный набор признаков, а как высокоорганизованное динамичное образование, интегрирующее результаты эмпирических наблюдений, теоретических обобщений и деятельностных компонентов. Оно обеспечивает управляемый переход обучающегося от анализа частных характеристик изучаемого объекта к синтезу его целостного многомерного образа в когнитивной структуре. Такое понятие всегда содержит несколько уровней абстракции – от конкретных примеров и операций до общих закономерностей и принципов.

Учитель информатики, работая с комплексными понятиями, должен не только передать знания, но и организовать учебную деятельность таким образом, чтобы ученик самостоятельно выделял существенные связи, сравнивал различные случаи и обобщал их в устойчивые мысленные конструкции. Теоретическим основанием для понимания природы этого процесса служат идеи Л. С. Выготского [5], который подчёркивал, что формирование подлинно научных понятий осуществляется исключительно через зону ближайшего развития, где социокультурные инструменты – алгоритмические конструкции, формальные языки, опосредованные формы учебной деятельности – играют ключевую роль. Именно их освоение обеспечивает переход от спонтанных, интуитивных представлений учащихся

к обобщениям, обладающим высокой степенью системности, доказательности и осознанности [5]. Так, например, житейское понятие «порядок действий» в быту может быть размытым, тогда как в информатике оно преобразуется в строгую алгоритмическую конструкцию «следование», где каждая команда выполняется последовательно, без пропусков и перестановок. Алгоритмизация деятельности учителя как раз и позволяет системно проводить такую трансформацию житейских представлений в научные понятия через специально спроектированные учебные ситуации.

После того как природа комплексного понятия и закономерности его формирования у обучающихся были раскрыты, необходимо обратиться к вопросу о том, какими конкретными средствами педагог может перевести эту теоретическую модель в диагностическую практику. Методологическую основу для операционализации работы учителя с комплексными понятиями образует ряд фундаментальных концепций отечественной и зарубежной дидактики. В числе наиболее значимых – теоретические разработки А. В. Усовой, посвящённые формированию научных понятий в учебном процессе. По Усовой, понятие выступает одновременно в трёх ипостасях: 1) как объект познания (то, что изучается), 2) как орудие познания (инструмент для анализа других явлений): 3) как конечный результат обучения (итоговое знание, остающееся у ученика). Исходя из этого, учитель информатики обязан проводить тщательный системный анализ как объёма понятия (совокупности

объектов, подпадающих под данное определение), так и его содержания (системы существенных инвариантных признаков). Только такой двойственный анализ гарантирует успешное усвоение знаний и его последующее практическое применение [20]. Применительно к информатике это означает, что понятие «структура данных», например, следует рассматривать и через объём (массивы, списки, стеки, очереди, деревья, хеш-таблицы), и через содержание (способы хранения элементов, операции доступа, инварианты упорядоченности). Алгоритмизируя свою деятельность, учитель составляет для каждого комплексного понятия так называемый «понятийный паспорт», фиксирующий объём, содержание, типичные примеры и контрпримеры, а также наиболее распространённые ошибки в трактовке.

Важным направлением в подготовке будущих учителей информатики, непосредственно связанным с освоением методов работы с комплексными понятиями, является обучение алгоритмизации и программированию. Методические особенности этого раздела школьного курса информатики рассматриваются в работах В. Ю. Нефедовой и К. К. Киякбаевой [14], а система учебных задач для формирования алгоритмической компетенции при подготовке учителей информатики разработана Е. В. Барановой и И. В. Симоновой [2]. Особенности подготовки будущих учителей информатики по разделу программирования – с акцентом на методическую составляющую работы с понятийным аппаратом – исследованы И. В. Левченко, А. Р.

Садыковой, Д. Б. Абушкиным, Л. И. Карташовой, В. А. Кондратьевой и В. П. Моисеевым [10]. Вычислительный аспект подготовки учителей информатики, в том числе освоение понятий вычислительного мышления, рассматривается в работе Г. Б. Камаловой [7].

Существенный вклад в разработку методологической базы анализа комплексных понятий вносит Н. И. Пак, предложивший оригинальную таксономию учебных целей с позиций ментального подхода. В данной концепции знание трактуется не как пассивный набор фактов, а как динамичная совокупность ментальных образов, схем и моделей, определяющих когнитивное поведение человека в цифровом обществе. Умения, в свою очередь, проявляются в способности гибко использовать эти внутренние структуры для снятия неопределённости при решении нестандартных задач. Данная ментальная таксономия органично интегрирует пересмотренные идеи таксономии Блума–Андерсона, выделяющей четыре типа знаний (фактологические, концептуальные, процедурные, метакогнитивные) и шесть уровней когнитивных процессов (помнить, понимать, применять, анализировать, оценивать, создавать).

Такой подход обосновывает алгоритмизацию как ключевой метакогнитивный инструмент, позволяющий учителю точно диагностировать, своевременно корректировать и непрерывно оптимизировать процесс

формирования комплексных понятий на всех уровнях когнитивной деятельности обучающихся [15].

Практическое применение ментальной таксономии можно проиллюстрировать на примере понятия «информационная модель». Алгоритмизированная последовательность работы учителя строится по уровням:

Уровень «помнить»: перечислить виды информационных моделей (табличные, иерархические, сетевые).

Уровень «понимать»: объяснить, почему табличная модель подходит для базы данных учеников.

Уровень «применять»: по описанию предметной области построить иерархическую модель.

Уровень «анализировать»: сравнить две модели одной системы и выявить преимущества и недостатки.

Уровень «оценивать»: выбрать оптимальную модель для заданной цели моделирования.

Уровень «создавать»: разработать собственную информационную модель для новой задачи.

Каждый из этих уровней предполагает специфические диагностические материалы и критерии оценки, а алгоритмизация обеспечивает полноту охвата всех уровней и возможность возврата к предыдущему этапу при обнаружении затруднений у учащихся.

Ключевую роль в процессе визуализации формирования комплексных понятий играют структурно-ментальные схемы, разработанные и обоснованные Е. В. Асауленко, И. В. Баженовой, М. М. Клунниковой и Н. И. Паком. Данные схемы выполняют функцию когнитивных инструментов, делающих видимыми скрытые процессы формирования, развития и оценки учебных компетенций, связанных с вычислительным мышлением. Авторы выделяют три основных типа учебных задач и соответствующих им структурно-ментальных схем [1].

Расчётные задачи, решаемые с помощью схем на основе вычислительных примитивов (арифметические и логические операции, сравнения). Данный тип задач формирует базовое понимание того, как из простых действий складываются сложные вычислительные процессы. Например, при изучении понятия «среднее арифметическое» учащиеся выстраивают ментальную схему: сложить все элементы \rightarrow подсчитать их количество \rightarrow разделить сумму на количество. Подобная схема служит прототипом любого вычислительного алгоритма и создаёт основу для осмысления более сложных структурных построений.

Операционные задачи, под которые разрабатываются иерархические процедурные схемы, позволяющие раскладывать сложные процессы на подпроцессы. В ходе их решения учащиеся осваивают приёмы выделения вспомогательных алгоритмов, передачи параметров и получения возвращаемых результатов. При изучении понятия «структурное

программирование» операционные задачи способствуют осмыслению принципа «разделяй и властвуй»: ментальная схема включает главный модуль и вызовы подпрограмм, наглядно отражая иерархию управления.

Алгоритмические задачи, которые представляются в классических блок-схемах с опорой на базовые алгоритмические конструкции: следование, ветвление, цикл. Данные задачи в наибольшей степени соответствуют традиционному пониманию алгоритмизации. Блок-схема здесь выступает внешней опорой для ментальной модели, позволяя ученикам отслеживать поток управления и состояние переменных на каждом этапе исполнения алгоритма. Дифференцированный подход к типам задач даёт учителю возможность алгоритмизировать анализ сложных понятий через поэтапное построение и применение ментальных моделей, обеспечивая плавный переход школьников от поверхностного, отрывочного восприятия к углублённому осмыслению, самостоятельной практической работе и рефлексии полученных результатов [1].

Следует подчеркнуть, что структурно-ментальные схемы всех трёх разновидностей допускают формализацию в графовом виде: вершины графа соответствуют концептуальным элементам понятия либо операциям алгоритма, а рёбра – смысловым, логическим или процедурным взаимосвязям между ними. Такое графовое представление не только упрощает машинную обработку схемы, но и задаёт единый формальный язык для описания

понятийной структуры, необходимый в дальнейшем для сопоставления усвоенных учеником знаний с эталонным образцом.

Описанный инструментарий (понятийный паспорт, ментальная таксономия Пака, структурно-ментальные схемы) образует целостную методологическую основу для алгоритмизации деятельности учителя информатики при диагностике сформированности комплексных понятий. Показательно, что каждый из перечисленных инструментов одновременно решает две задачи: с одной стороны, он структурирует работу самого педагога, придавая ей системный и воспроизводимый характер; с другой – создаёт формализованное описание педагогических действий, пригодное для последующей реализации в архитектуре интеллектуального агента. В этом смысле автоматизация профессиональной деятельности учителя является не просто теоретическим упражнением, но необходимым условием разработки работоспособной агентной системы поддержки педагогической диагностики.

Таким образом, теоретико-методологический анализ деятельности учителя информатики при работе с комплексными понятиями позволяет выделить несколько уровней её организации:

Уровни организации деятельности учителя.

Уровень	Содержание и дидактическая функция
Операциональный	организация учебной деятельности по поэтапному освоению понятия с опорой на ментальные схемы
Оценочный	соотнесение достигнутых результатов с требованиями таксономии учебных целей
Диагностический	определение исходного уровня сформированности понятийного аппарата
Структурирующий	выделение объёма и содержания понятия, построение понятийного паспорта

Каждый из этих уровней поддаётся формализации в виде алгоритма с условиями и разветвлениями, что открывает возможность его воспроизведения в рамках интеллектуальной агентной системы. Именно этот потенциал формализуемости педагогических действий составляет теоретическую базу для перехода к практическому проектированию ИИ-агента.

Весь охарактеризованный инструментарий задают разные срезы анализа комплексного понятия, однако каждый из них в конечном счёте требует единого, пригодного для машинной обработки представления результата. Именно в таком представлении выступает граф понятий, построение которого замыкает всю описанную методологическую работу учителя и образует завершающий этап её автоматизации.

Наиболее ресурсоёмким этапом в деятельности учителя при диагностике сформированности комплексных понятий является построение двух взаимосвязанных граф-моделей: эталонного семантического графа понятия, выстроенного на основе нормативных требований образовательных стандартов, и семантического графа предполагаемого усвоения понятия конкретным учеником, воссоздаваемого по результатам диагностических процедур. Первый граф представляет собой идеальный образец структуры знания: он фиксирует совокупность ключевых концептуальных узлов, соответствующих существенным признакам понятия, и типы связей между ними: родовидовые, причинно-следственные, операциональные, атрибутивные. Учитель строит эталонный концептуальный граф на базе нормативной документации: федеральных государственных образовательных стандартов, рабочих программ и материалов учебников [9]. В противовес ему существует индивидуальный граф, который фиксирует реальную когнитивную структуру учащегося: в нём отражены уже усвоенные понятийные узлы, сложившиеся между ними взаимосвязи, а также зоны неопределённости, смысловые искажения или внепрограммные (избыточные) элементы, не соответствующие нормативной модели.

Сравнение этих двух графов выводит педагога за рамки субъективных, чисто качественных представлений о том, насколько сформировано то или иное понятие, и даёт возможность получить формализованную, воспроизводимую оценку, допускающую численное выражение. В ходе такого

анализа становятся очевидными как лакуны в понятийном аппарате ученика (недостающие вершины и рёбра), так и закреплённые ошибочные конструкции, которые требуют адресной корректирующей работы. Именно эта операция – синтез нормативного и индивидуального представления комплексного понятия в виде графа составляет ключевой диагностический шаг в деятельности учителя и является центральным объектом нашего исследования.

Вместе с тем данный этап представляет собой один из наиболее трудозатратных в практике учителя: построение эталонного графа требует аналитической работы со значительным массивом нормативных и методических текстов, нередко представленных в разных форматах – печатных документах, электронных таблицах, методических пособиях, мультимедийных материалах [3]. Реконструкция же индивидуального семантического графа ученика предполагает интерпретацию его развёрнутых письменных и устных ответов, концептуальных карт, схем и иных продуктов учебной деятельности, также имеющих мультимодальный характер. Именно здесь возможности интеллектуального агента, способного извлекать, структурировать и сопоставлять информацию из мультимодальных источников (текстовых, графических, табличных, звуковых и т.д.) приобретают принципиальное значение. Агент, оснащённый инструментами обработки мультимодального контента, способен автоматизировать как построение эталонного графа по совокупности нормативных документов, так

и формирование семантического графа усвоения понятий учеником, на основе анализа диагностических материалов учащегося.

Таким образом, построение семантических графов (эталонного и индивидуального) выступает тем связующим звеном, которое соединяет теоретико-методологический анализ деятельности учителя, представленный в данном параграфе, с практической задачей проектирования интеллектуальной агентной системы педагогической диагностики. Передача этого этапа агенту не означает устранения педагога из диагностического процесса: учитель сохраняет за собой функции целеполагания, интерпретации результатов и принятия педагогических решений [6]. Однако рутинная и ресурсоемкая работа по извлечению структуры знания из разнородных источников и её формализации в виде графовых моделей становится зоной эффективного человеко-машинного взаимодействия, в которой ИИ-агент способен существенно разгрузить учителя и повысить объективность диагностики.

1.3 Обоснование архитектуры и функциональных требований к ИИ-агенту в контексте профессиональной деятельности педагога: теоретико-методологический и технологический аспекты

С 2023 года искусственный интеллект превратился в технологию, активно трансформирующую экономики ведущих стран мира. Соответствующие продукты, известные как большие языковые модели, доступны в виде чат-ботов ряда крупнейших разработчиков: ChatGPT от компании OpenAI, Claude от компании Anthropic, а также их конкурентов – DeepSeek, Qwen и GigaChat. Эти генеративные нейронные сети способны создавать текст на произвольную заданную тему – от исходного кода программы до художественных произведений. В настоящее время подобные инструменты приобретают мультимодальный характер, то есть становятся способны принимать на вход и генерировать не только текст, но и изображения, звук, файлы формата PDF и т.д. [33].

Подобная трансформация обусловлена тем, что модели способны понимать запрос, сформулированный на естественном языке, и предоставлять качественный ответ на уровне, сопоставимом с уровнем квалификации кандидата наук. Это привело к тому, что любой пользователь получает возможность получить квалифицированный и персонализированный прямой ответ – результат, ранее недостижимый средствами традиционных поисковых систем. Модель берёт на себя всю когнитивную работу: поиск возможных источников информации для ответа, понимание сути вопроса, его переформулирование на язык предметной области, получение ответа путём

вывода из источников и перевод ответа обратно на доступный пользователю язык.

Обычный чат-бот способен принимать на вход источник достоверной информации (PDF-файл, стандарт или регламент) и запрос (так называемый промт), формируя на их основе ответ на любой вопрос, связанный с содержанием источника. Благодаря этому модель способна применять современные наработки ведущих методистов, представленных непосредственно в виде статей, сразу в учебном процессе: генерировать планы уроков, методические рекомендации, дидактические материалы, презентации, тренажёры и т. д. Тем самым учитель освобождается от необходимости самостоятельно глубоко перерабатывать научные статьи.

При работе с чат-ботами педагог сталкивается с необходимостью самостоятельно интерпретировать, верифицировать и адаптировать получаемые результаты под конкретные учебные цели, возрастные особенности обучающихся и требования федеральных государственных образовательных стандартов (ФГОС), даже если для этого достаточно лишь скорректировать промт и добавить в контекст диалога соответствующие документы (учебники, стандарты, расписание и т.д.). Методика и методология подобной работы в настоящее время только исследуется педагогическим сообществом.

Тем не менее, большие языковые модели (LLM) обладают значительными преимуществами, которые делают их привлекательным

инструментом для педагога. Крупные языковые модели (LLM) демонстрируют высокую генеративную производительность и способны эффективно обрабатывать большие объёмы информации, формируя по ним обоснованные выводы за относительно короткий срок [38]. Это позволяет оперативно анализировать архивы работ учащихся, учебные материалы и диагностические данные, генерировать варианты объяснений, упражнений и индивидуальных рекомендаций.

Однако, несмотря на указанные достоинства, базовые генеративные модели обладают рядом врождённых недостатков, принципиально ограничивающих их дидактическую ценность [38]. Их применение в педагогическом процессе сталкивается с системными ограничениями, в самой природе статистического предобучения. Результаты по умолчанию недостоверны, поскольку модель генерирует текст в соответствии с вероятностными паттернами, извлечёнными из обширного, но неструктурированного корпуса данных, а не на основе верифицированных педагогических критериев, дидактических принципов или предметных онтологий [31, 38].

Это приводит к нескольким критическим последствиям. Во-первых, феномену «галлюцинаций» генерации правдоподобного, но фактически неверного или вымышленного содержания, что особенно опасно в образовательном контексте, где точность и научная достоверность выступают базовыми требованиями [38]. Во-вторых, наблюдается нарушение логической

связности и внутренней когерентности выводов, поскольку модель не обладает устойчивым механизмом причинно-следственного мышления, а опирается на поверхностные статистические ассоциации. В-третьих, отсутствует прозрачность внутренних механизмов вывода: учитель не имеет доступа к цепочке промежуточных рассуждений, что делает невозможной оперативную верификацию и коррекцию результатов [31, 33].

Модель, по сути, «пишет так, как она пишет», не гарантируя соответствия возрастным психофизиологическим особенностям обучающихся, логике конкретной предметной области или требованиям федеральных государственных образовательных стандартов (ФГОС). Генерируемый материал может игнорировать зону ближайшего развития, не учитывать индивидуальные когнитивные стили или вступать в противоречие с принципами системности и операциональности знаний. В результате учитель вынужден постоянно выступать в роли корректора, интерпретатора и финального редактора: проверять фактическую точность, адаптировать текст под возрастные нормы, устранять методические несоответствия и обеспечивать дидактическую целесообразность. Такая дополнительная деятельность существенно повышает когнитивную и временную нагрузку на педагога, приводит к быстрому профессиональному выгоранию и, как следствие, снижает общую эффективность внедрения ИИ в образовательную практику.

Эпизодическое использование LLM не позволяет достичь системной интеграции искусственного интеллекта в педагогику. Переход к более зрелым архитектурам, в частности, к агентным системам становится необходимым условием, обеспечивающим преодоление указанных ограничений через механизмы структурированного планирования, долговременной памяти, внешних инструментов и runtime-governance.

Агентные системы (agentic AI) представляют собой надстройки над LLM [31]. Работа ИИ-агентов строится по следующему циклу:

- получают от пользователя описание желаемого результата,
- самостоятельно обращаются к LLM для генерации шагов, необходимых для достижения результата (это становится возможным только благодаря мощности современных LLM, аккумулирующих огромный массив практических кейсов),
- извлекают из ответа LLM план действий в виде набора конкретных шагов
- преобразуют шаги плана в вызовы инструментов и действия, которые необходимо выполнить на компьютере пользователя
- *выполняют их от имени пользователя*
- анализируют результаты выполнения на предмет ошибок
- исправляют выявленные ошибки, корректируя выдвинутые гипотезы и план действий

Цикл повторяется до тех пор, пока языковая модель не подтвердит выполнение поставленной задачи.

Ключевое отличие от LLM заключается в наличии обратной связи. Например, если спросить, как запустить файл «main.py», чат-бот лишь подскажет выполнить команду «python main.py» и даст ряд рекомендаций. Агент же самостоятельно выполнит эту команду, и если получит ошибку «Python не установлен в системе», сформулирует ряд гипотез (Python отсутствует, он не указан в переменной PATH, ...), проверит каждую из них и на основании результатов проверки сможет либо запустить файл уточнённой командой «C:\Python\python.exe main.py», либо скачать и установить Python и с помощью только что установленного интерпретатора выполнить скрипт. Важно отметить, что агент действует активно и не останавливается до достижения поставленной цели, а благодаря возможностям современных LLM обладает обширными знаниями об операционных системах, языках программирования, способах настройки и т.д.

Говорят, что агент представляет собой goal-oriented [22] автономную вычислительную сущность, соединяющую рассуждение и действие посредством моделей большого объёма, систем памяти и внешних инструментов для достижения контекстно-обусловленных результатов [31, 33]. В отличие от статичных генеративных моделей, агентные системы демонстрируют автономию, проактивность, память, само-рефлексию и координацию с другими агентами.

Во-первых, агенты минимизируют goal drift (смещение целей) за счёт структурированного планирования, итеративного уточнения и механизмов runtime-governance [30].

Одним из наиболее значимых ограничений LLM в педагогическом применении является склонность к неконтролируемому отклонению от исходной задачи, явлению, обозначаемому как goal drift [22] (смещение целей). В педагогическом контексте это приводит к критическим рискам: агент, генерирующий рабочую программу, может, например, постепенно отклониться от требований Федерального государственного образовательного стандарта (ФГОС), что потребует от учителя тотальной, пошаговой верификации каждого сгенерированного блока. Агентные системы преодолевают данную проблему за счёт принципиально иной архитектуры.

Структурированное планирование позволяет агенту на начальном этапе декомпозировать сложную педагогическую задачу на последовательность подцелей, явно фиксируя их в виде промежуточных состояний и критериев достижения. Итеративное уточнение (iterative refinement) [40] обеспечивает регулярный пересмотр текущего плана с учётом новых данных, обратной связи от учителя или результатов промежуточных действий. Наконец, механизмы runtime-governance такие как goal-conditioned drift detection [22], goal-aware authorization monitoring [22] и finite-state conformance engines – осуществляют непрерывный мониторинг поведенческих последовательностей агента в реальном времени.

Центральное место в этой системе занимает механизм goal-conditioned drift detection [22], который сопоставляет текущее поведение агента с эталонной целевой моделью и фиксирует отклонения, выходящие за рамки нормальной адаптации и сигнализирующие о риске потери фокуса. Благодаря этому агент сохраняет операционную устойчивость даже при динамичных изменениях образовательного контекста (например, когда в ходе работы над индивидуальным маршрутом ученика корректируется диагностическая цель). Снижение goal drift [22] превращает ИИ-агента из непредсказуемого генератора в прогнозируемый и методологически корректный инструмент поддержки, что укрепляет доверие педагога и упрощает интеграцию ИИ в повседневную практику.

Дополнительные гарантии управляемости и безопасности обеспечиваются за счёт guardrails [25, 27] (отклоняющих или исправляющих нежелательные ответы), цепочек рассуждений Chain-of-Thought [39] и специализированных систем мониторинга поведенческих последовательностей [31, 38]. Помимо этого, сложные педагогические задачи разбиваются на этапы с обязательным участием учителя (human-in-the-loop) перед отправкой финального запроса в LLM, что гарантирует практическую применимость и проверяемость результатов.

Среди конкретных реализаций подобных решений выделяются локальные агентные платформы: OpenCode [36], AutoGPT [23] и CrewAI [26].

AutoGPT [23], созданный Тораном Брюсом Ричардсом, исторически является одной из первых автономных реализаций на базе LLM, работающей в цикле ReAct. Это делает его удобным инструментом для генерации несложных методических заготовок, однако ограничивает его применение в многокомпонентных задачах, требующих анализа больших массивов данных или междисциплинарного синтеза.

CrewAI [26] (Жоан Моура) специализируется на ролевом взаимодействии агентов и реализации кооперативных паттернов, что теоретически привлекательно для педагогических коллективов. Однако на практике он требует сложной настройки, привязан к внешним сервисам и страдает от непрозрачности логических выводов, что делает его неприемлемым в образовании, где важна ясность каждого шага.

OpenCode выгодно отличается полной открытостью, простотой локального развёртывания и модульной системой навыков. Это предоставляет образовательным учреждениям полный контроль над процессами, независимость от облачных провайдеров и максимальную прозрачность, что становится решающим доводом в пользу его выбора. Кроме того, как open-source проект, OpenCode не подвержен санкционным ограничениям в отношении РФ, что обеспечивает его доступность и возможность использования в отечественной школе без риска внезапного прекращения поддержки или потери доступа.

В отличие от облачных чат-ботов, которые требуют передачи контекста в каждом сеансе, OpenCode работает локально в рабочей папке, содержащей все необходимые для педагогической деятельности материалы: учебные планы, рабочие тетради, результаты диагностик, цифровые следы уроков и архивы предыдущих взаимодействий. Агент извлекает данные из файлов через специализированные инструменты (skills), обрабатывает их, генерирует новые артефакты и тем самым формирует устойчивую долговременную память. Это позволяет создать информационный слепок учебного процесса – динамическую репрезентацию когнитивного развития как каждого отдельного учащегося, так и класса в целом.

Благодаря локальному запуску и инструментальной архитектуре педагог получает инструменты опосредованной работы с накопленными данными: конвертация в требуемый формат осуществляется с минимальными затратами времени и сил. Учитель может запросить у агента «воспоминание» о предыдущих уроках, провести аналитику динамики освоения компетенций, выполнить прогностическую оценку рисков отставания или получить персонализированные рекомендации по корректировке траектории. На передний план выходит диагностический аспект: система загружает исходный файл с ученической работой, формирует эталонную модель понятий согласно требованиям ФГОС (в виде структурированного дерева знаний), сравнивает её с персональной моделью учащегося и отображает обнаруженные расхождения в формате двух иерархических деревьев. Пробелы оцениваются по параметрам

глубины, взаимосвязанности и применимости понятий, что даёт педагогу возможность не просто констатировать «неусвоенный материал», но и выявить системные корни этих недочётов.

К числу ключевых свойств относится абсолютная открытость логики принятия решений. Цепочка рассуждений (Chain-of-Thought) представлена в читаемом виде: преподаватель имеет доступ к полному протоколу всех промежуточных этапов, может проверить обоснованность выводов, предложить собственное педагогическое толкование и инициировать пересчёт результата. Благодаря этому OpenCode превращает фрагментированные данные в действенную педагогическую базу, при этом безусловный приоритет остаётся за человеком.

Из этого следует, что переход на агентные системы предпочтителен, поскольку обеспечивает четыре ключевых преимущества: наличие специализированных умений (skills), повышенную управляемость, прозрачность рассуждений и устойчивый контекст. Именно эти умения и составляют основу электронного учителя - набора педагогических функций, реализованных в форме инструментальных модулей. К ним относятся: автоматизированная диагностика по ФГОС, генерация индивидуальных траекторий, анализ динамики когнитивных структур, формирование адаптивных дидактических материалов. Такие модули подготовлены и находятся по ссылке на Яндекс Диске.

В более широком контексте внедрение агентных архитектур отвечает фундаментальным вызовам современной педагогики. В условиях роста информационной энтропии и индивидуализации обучения традиционные методы организации учебного процесса демонстрируют ограничения по масштабируемости и персонализации. Интегрируя принципы системности и операциональности знания, агентные системы обеспечивают переход от фрагментарного, эпизодического применения ИИ к непрерывному системному сопровождению образовательного процесса [25, 8]. При этом приоритет человеческого фактора остаётся неизменным: педагог получает инструментарий для рефлексии, диагностики и проектирования, а ученик – условия для становления субъектной позиции и развития метакогнитивных способностей.

Вместе с тем ключевое значение приобретают архитектурные решения, направленные на безопасность и этичность функционирования. Механизмы управления в реальном времени (runtime-governance) – такие как индикатор агентных рисков (Agency-Risk Index), целенаправленная авторизация (goal-aware authorization) и градуированное ограничение полномочий (graded containment) – позволяют свести к минимуму угрозы дрейфа целей и неконтролируемых действий [38]. В приложении к педагогике это означает, что любые рекомендации, исходящие от агента, подвергаются проверке на соответствие возрастным рамкам, этическим нормам и заявленным образовательным задачам. Педагогическая методология здесь требует не

только технического воплощения, но и органичного сопряжения с психологическими закономерностями интериоризации, учётом зоны ближайшего развития и принципами совместной деятельности.

Особое внимание в контексте педагогического применения агентных систем следует уделить вопросам безопасности и предотвращения утечек персональных данных. С позиции российского законодательства обработка персональных данных обучающихся, их родителей и педагогов регулируется Федеральным законом от 27.07.2006 № 152-ФЗ «О персональных данных» (с изменениями), который устанавливает строгие требования к конфиденциальности, целесообразности, минимизации объёма обрабатываемых данных и обеспечению их защиты. В образовательных организациях, выступающих операторами персональных данных, передача сведений, относящихся к категории специальных или биометрических (включая результаты диагностик, индивидуальные траектории, когнитивные профили), на внешние облачные серверы без надлежащих гарантий создаёт высокие правовые риски: от административной ответственности (штрафы по КоАП РФ) до уголовной ответственности (ст. 137 УК РФ и ст. 272.1 УК РФ).

Облачные генеративные модели и удалённые LLM-провайдеры неизбежно предполагают внешнюю передачу данных, что вступает в противоречие с требованиями локализации и контроля. Даже при использовании соглашений о неразглашении сохраняется вероятность утечек, инцидентов на стороне провайдера или несанкционированного доступа. В

таких условиях обеспечение соответствия законодательству становится крайне затруднительным.

Обеспечение безопасности и правовой безупречности при обработке персональных данных в агентных системах становится возможным исключительно в случае применения локальных LLM через Ollama (либо аналогичные инструменты). Локальное развёртывание гарантирует полную изоляцию сведений: все стадии восприятия, логического вывода и генерации проходят на устройстве учителя или школьном сервере, без передачи информации во внешнюю среду. Такой подход полностью исключает риски утечки данных на этапе взаимодействия с моделью, соответствует положениям ФЗ-152 в части защиты персональных данных в информационных системах (включая мероприятия по предотвращению несанкционированного доступа и размещение данных в контролируемом пространстве), а также даёт образовательной организации возможность сохранять за собой статус оператора, который осуществляет всесторонний контроль на каждом этапе обработки данных.

Стоит подчеркнуть, что внедрение агентных систем в педагогическую практику – это не технологическая неизбежность, а методологический выбор. Он предполагает пересмотр роли учителя: от транслятора готовых знаний к проектировщику образовательной среды и координатору индивидуального развития учащихся. При сохранении приоритета человеческого контроля и опоре на проверенные педагогические теории подобные инструменты

способны значительно повысить эффективность, персонализацию и развивающий потенциал образовательного процесса.

В заключение отметим, что внедрение агентных систем в педагогическую практику не является технологической данностью, а представляет собой методологический выбор. Оно требует переосмысления роли учителя - от транслятора знаний к проектировщику образовательных сред и координатору развития. При условии сохранения приоритета человеческого контроля и опоры на проверенные педагогические теории такие системы способны существенно повысить эффективность, индивидуальность и развивающий потенциал образования. Готовые навыки (skills) электронного учителя, размещённые в доступных репозиториях, открывают путь к широкому практическому внедрению уже сегодня.

Вывод по первой главе

В первой главе выпускной квалификационной работы обоснованы теоретико-методологические основы применения ИИ-агентов для анализа сформированности комплексных понятий в школьном курсе информатики. Установлено, что в условиях цифровой трансформации образования ИИ-агент выступает не как средство элементарной автоматизации, а как полноценный когнитивный посредник, расширяющий операциональную сферу педагогической деятельности. Опираясь на деятельностный подход А. Н. Леонтьева [12] и культурно-историческую теорию Л. С. Выготского [5], методология применения таких систем предполагает их глубокую интеграцию в структуру профессиональной деятельности учителя. Это обеспечивает динамический мониторинг формирования понятийных структур в зоне ближайшего развития при строгом соблюдении принципов прозрачности, объяснимости и этической безопасности в соответствии с рамками NIST AI RMF 1.0 и стандартами ISO/IEC.

В развитие данного положения доказана объективная необходимость алгоритмизации деятельности учителя при работе с комплексными понятиями. Синтезируя диалектическую концепцию научного понятия А. В. Усовой и ментальный подход Н. И. Пака, алгоритмизация действий педагога рассматривается как универсальный педагогический метаинструмент. Она обеспечивает целенаправленный переход от диагностики житейских представлений учащихся к критериальному оцениванию и своевременной коррекции учебных действий. Использование структурно-ментальных схем и

технологических карт позволяет снизить непродуктивную когнитивную нагрузку и гарантировать системность формирования вычислительного мышления обучающихся.

На основе выявленных методологических потребностей сформулированы и научно обоснованы функциональные и архитектурные требования к ИИ-агенту. Показано, что базовые генеративные модели обладают рядом врождённых ограничений, склонностью к «галлюцинациям», нарушением логической связности выводов и отсутствием прозрачности механизма вывода, что делает эпизодическое использование LLM недостаточным для системной педагогической диагностики и обуславливает необходимость перехода к агентным архитектурам, обеспечивающим структурированное планирование, минимизацию дрейфа целей (goal drift) и механизмы runtime-governance. Сопоставление AutoGPT, CrewAI и OpenCode показало, что последний выгодно отличается модульной архитектурой с приоритетом конфиденциальности. Выбор в пользу OpenCode обусловлен несколькими ключевыми факторами: обеспечением цифрового суверенитета над данными, возможностью локального запуска через Ollama, встроенной поддержкой паттернов RAG, Chain-of-Thought и ReAct, а также наличием механизмов операционного контроля во время исполнения. Дополнительным доводом в пользу локального способа развёртывания служат нормы Федерального закона от 27.07.2006 № 152-ФЗ «О персональных данных»: проведение диагностических процедур и анализ когнитивных профилей

учеников исключительно в закрытом внутреннем контуре, без какого-либо внешнего обмена, позволяет свести юридические риски к минимуму и обеспечивает неукоснительное соблюдение принципа «учитель в контуре» (teacher in the loop).

Таким образом, в первой главе выстраивается целостная теоретическая конструкция, в которой методология применения ИИ-агентов, алгоритмизация педагогической деятельности (в том числе операция сличения эталонного и индивидуального семантических графов понятий), а также жёсткие архитектурные и правовые требования к самому агенту находятся в отношениях взаимного обусловливания и дополнения. Полученные на этом этапе выводы образуют научно обоснованный фундамент для перехода ко второй главе, посвящённой проектированию, программной реализации и экспериментальной проверке ИИ-агента, предназначенного для автоматизированного оценивания степени сформированности сложных понятий.

Глава 2. Проектирование и апробация программного решения для автоматизации диагностики знаний

2.1. Разработка системы команд и навыков ИИ-агента для диагностики сформированности понятий школьного курса информатики

В данном параграфе рассматривается процесс проектирования системы команд и внутренних навыков, специализированного ИИ-агента. В развитие положений разделов 1.1-1.3, опираясь на деятельностный подход А.Н. Леонтьева [12] и культурно-историческую теорию Л.С. Выготского [5], агент концептуализируется как орудие, опосредующее диагностическую деятельность педагога, расширяющее его операциональную сферу и обеспечивающее эффективную работу в зоне ближайшего развития обучающихся [11; 18]. Такое понимание позволяет перейти от традиционных, преимущественно интуитивных методов контроля к системному, структурированному формирующему оцениванию, полностью соответствующему задачам цифровой трансформации школьного образования.

Одной из наиболее острых педагогических проблем современного школьного курса информатики остается недостаточная точность и глубина диагностики сформированности комплексных понятий. Учитель, опираясь преимущественно на итоговые ответы учащихся в виде готовых формулировок, фрагментов кода или схем, зачастую не в состоянии выявить точную локализацию пробела в структуре знания. Дефицит может быть связан с непониманием ключевых атрибутов понятия, нарушением системных связей между его компонентами, отсутствием способности к операционализации или

переносу в новые контексты, а также с фрагментарностью ментальных схем, формируемых в процессе обучения. Традиционные формы контроля тестовые задания, устные опросы, проверка практических работ преимущественно фиксируют поверхностный, репродуктивный уровень усвоения, оставляя скрытыми внутренние механизмы формирования понятийных структур. В связи с этим возникает объективная потребность в инструменте, способном анализировать не только правильность конечного ответа, но и глубинную архитектуру знания, включая иерархию понятий, типы связей (род-вид, часть-целое, функциональная, ассоциативная), степень интериоризации и соответствие таксономии учебных целей по Н.И. Паку. Именно такую диагностическую функцию выполняет проектируемый ИИ-агент, выступающий в роли когнитивного ассистента педагога, который, в развитие идей главы 1, позволяет алгоритмизировать процесс выявления и коррекции пробелов в понятийном аппарате обучающихся.

Общая характеристика ИИ-агента определяется его глубокой ориентацией на реальные условия образовательного процесса в строгом соответствии с логикой Федерального государственного образовательного стандарта (ФГОС) общего образования. Агент интегрируется в контур профессиональной деятельности учителя информатики как полноценный когнитивный посредник, обеспечивающий переход от статичной, суммативной диагностики к динамическому формирующему оцениванию, основанному на непрерывном мониторинге развития понятийных

образований. В отличие от универсальных генеративных моделей, агент реализует специализированные педагогические сценарии, опираясь на методологию, обоснованную в первой главе: алгоритмизацию деятельности учителя (раздел 1.2), ментальный подход Н.И. Пака, структурно-ментальные схемы Е.В. Асауленко [1] и соавторов, а также принципы чистой архитектуры и *runtime governance*, изложенные в разделе 1.3 при обосновании выбора *Opencode*. Это позволяет агенту выступать не в роли замены педагога, а в качестве инструмента, усиливающего его рефлексивную, проектировочную и аналитическую компоненты деятельности, минимизируя при этом непродуктивную когнитивную нагрузку и повышая точность педагогических решений в условиях ограниченного учебного времени.

Система команд, доступных учителю при взаимодействии с агентом, структурирована в четыре взаимосвязанных блока: настройку контекста, диагностику знаний, аналитику результатов и сопровождение обучения, каждый из которых содержит набор соответствующих умений (*skills*).

Блок настройки контекста обеспечивает первичную адаптацию агента к конкретным условиям педагогического процесса: учитель задает тему урока, параллель или конкретный класс, целевое комплексное понятие (например, «алгоритм», «структура данных», «информационная модель», «рекурсия»), а также загружает релевантные дидактические материалы (отрывки из рабочей программы, учебника, ФГОС или методических рекомендаций). Это позволяет агенту сформировать актуальную эталонную модель понятия с

учетом возрастных особенностей обучающихся и требований образовательного стандарта. Блок настройки контекста относится к умению `concept-decomposition-informatics`.

Блок диагностики знаний предполагает ввод ответа учащегося в различных форматах (текстовом описании, блок-схеме, фрагменте программного кода) и запуск многоуровневого семантического анализа на основе декомпозиции целевого понятия. Здесь агент сопоставляет представленное знание с эталонной структурой, выявляя степень соответствия по ключевым атрибутам, связям и уровням таксономии. Блок диагностики знаний реализуется посредством умения `diagnostic-response-analysis`.

Блок аналитики результатов включает генерацию детализированного отчета с указанием уровня сформированности понятия, визуализацией пробелов в виде семантического графа, классификацией дефицитов и формулировкой педагогически обоснованных выводов. Наконец, блок сопровождения обучения обеспечивает переход от диагностики к коррекции: агент автоматически подбирает или генерирует дидактические материалы - расчетные, операционные и алгоритмические задачи в соответствии со структурно-ментальными схемами, -ориентированные на устранение выявленных пробелов с учетом индивидуальной зоны ближайшего развития ученика. Блок аналитики результатов реализуется посредством скилла `concept-formation-tracker`.

Такая модульная организация команд полностью соответствует принципам алгоритмизации педагогической деятельности, подробно рассмотренным в разделе 1.2, и обеспечивает последовательное, рефлексивное управление процессом формирования понятий, минимизируя временные и когнитивные затраты учителя.

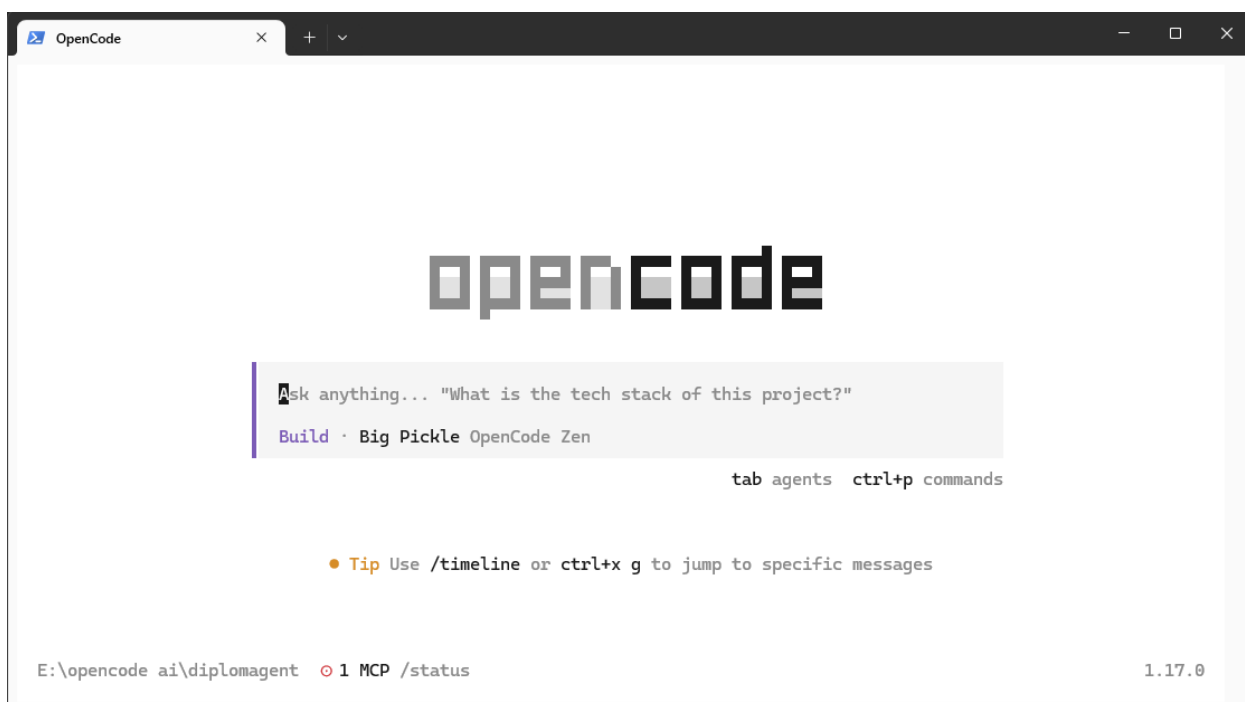


Рисунок 1 – Начальное окно агента

В блоке настройки контекста основным умением является concept-decomposition-informatics, который реализующий строгий алгоритм декомпозиции комплексного понятия. Этот алгоритм включает в себя семь последовательных этапов. На этапе подготовки контекста агент анализирует входные данные, находящиеся в папке «Дидактические материалы» (целевое понятие и релевантные отрывки из учебников, ФГОС, рабочих программ), формирует операциональное определение и выделяет ключевые аспекты.

Первичная декомпозиция предполагает формулирование четкого определения, выделение атрибутов, составление списка связанных понятий с указанием типа связи и уровня сложности. На этапе итеративного уточнения проводится мысленная проверка полноты, корректности типов связей и соответствия школьному курсу информатики.

Этапы построения семантического графа составляют ядро навыка и включают: 1) формирование логической модели, где каждая вершина представляет понятие с атрибутами (определение, уровень сложности), а каждое ребро -тип связи с направлением от более общего к частному или от целого к части; 2) валидацию и минимальную коррекцию, в ходе которой проверяется отсутствие изолированных вершин, циклических противоречий и несоответствий в уровнях сложности; 3) формирование итогового JSON-представления, содержащего nodes и edges; 4) визуализации изображение в формате PNG.

Умение *concept-decomposition-informatics* визуализирует семантический граф с помощью Mermaid [21]. Данный инструмент (интегрированный через `mcr-mermaid` и `mermaid-cli`) выступает как когнитивно-ориентированное средство внешней репрезентации ментальных структур. На завершающем этапе агент формирует Mermaid-код в нотации `graph LR` или `graph TD`, где узлы отображаются в формате `id["название (уровень)"]`, а ребра содержат подписи типов связей (`->/relation_type/`). Генерация PNG-файла с прозрачным фоном и высокими размерами (2400×1800) обеспечивает педагогически эффективную

визуализацию: учитель и ученик получают наглядный, иерархически организованный образ понятия, способствующий интериоризации системных связей. Mermaid обеспечивает доступность, масштабируемость и интеграцию с образовательными материалами, превращая абстрактный семантический граф в инструмент формирования целостного ментального образа в соответствии с подходом Н.И. Пака.

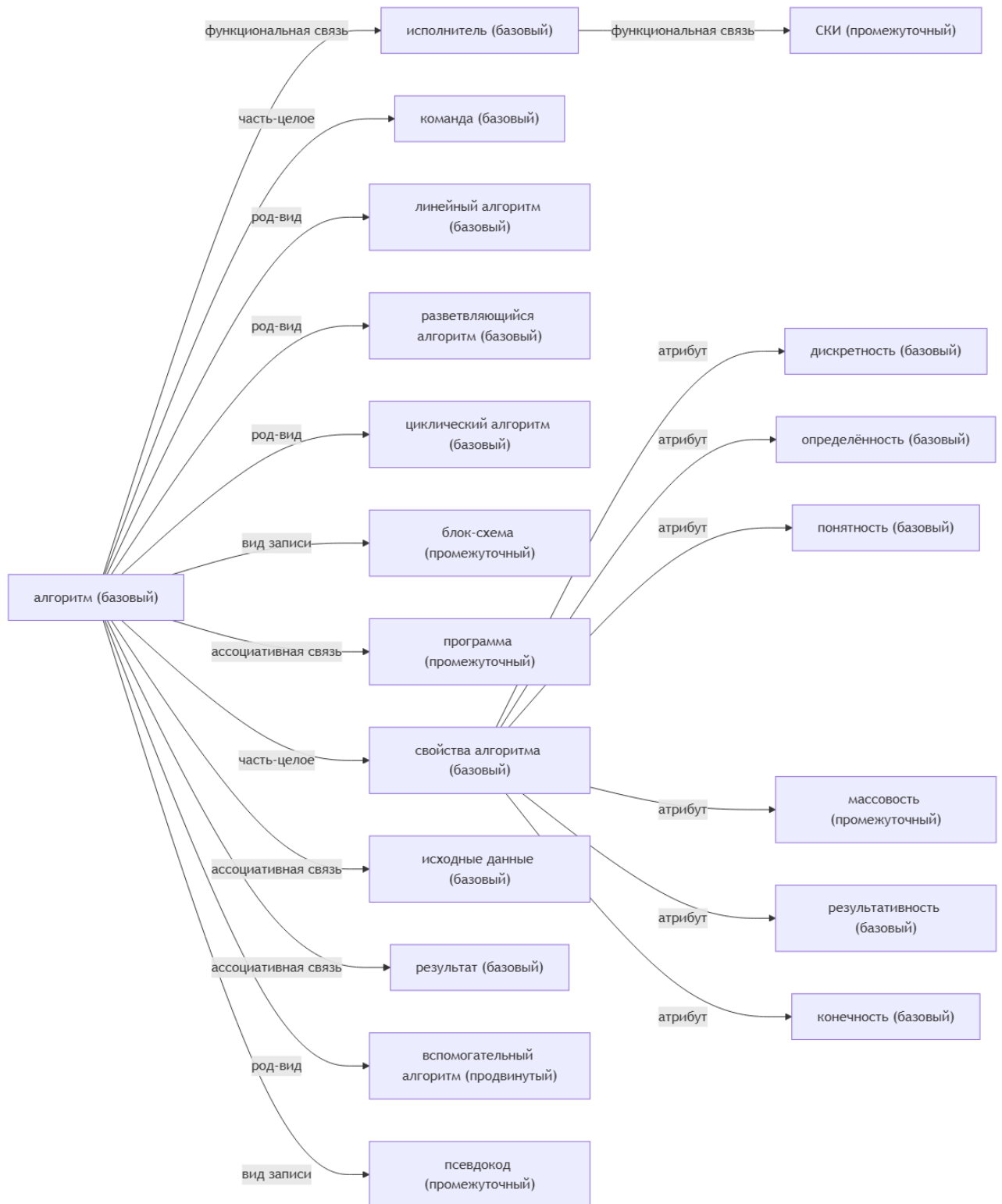


Рисунок 2 – Пример семантического графа понятия «Алгоритм» построенный с помощью вызова умения concept-decomposition-informatics

Для углубленной оценки индивидуальных ответов задействуется умение diagnostic-response-analysis. Он обеспечивает автоматизированный многоуровневый анализ с опорой на эталонную декомпозицию, оценивая степень сформированности понятия по таксономии Н. И. Пака (фактологический, концептуальный, процедурный, метакогнитивный, творческий уровни).

Сводка по 5 ученикам:

Ученик	Мак уровень	Статус	Ключевой вывод
student_01	концептуальный	присутствует	Знает виды и свойства, путает «точность» и «определённость», не понимает вспомогательный алгоритм
student_02	фактологический	частично	Грубые орфографические ошибки, знает 2 свойства из 6, требует индивидуальной коррекции
student_03	процедурный	присутствует	Точное определение, пример алгоритма площади, адекватная самооценка
student_04	творческий	выражено	Аналогии CPU/ISA, выход на проблему останова Тьюринга, требуется углубление
student_05	фактологический	частично	Одна фраза, полная неполнота, требуется повторение с нуля

Групповая статистика:

- Фактологический: 40% отсутствует/частично, 60% присутствует/выражено
- Творческий: **80% отсутствует** – самое слабое место группы
- Типичные ошибки: терминологические (40%), неполнота (40%)

Рекомендации учителю:

1. **Группа риска** (s02, s05) – повторение терминологии
2. **Продвинутая группа** (s04) – углублённый материал по теории алгоритмов
3. Фронтально: разбор 6 свойств, практикум по составлению алгоритмов, введение вспомогательного алгоритма

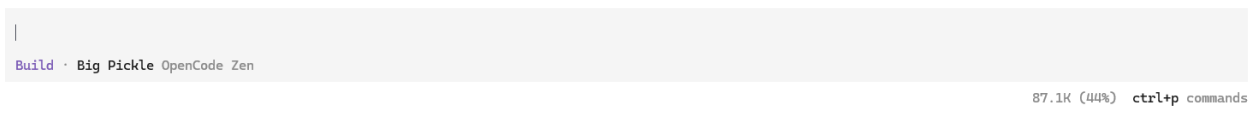


Рисунок 3 – Пример результата работы умения diagnostic-response-analysis

Долгосрочное отслеживание динамики формирования понятия реализуется через скилл concept-formation-tracker. Этот инструмент аккумулирует результаты серий диагностик в специализированном хранилище, строит индивидуальные и групповые траектории развития, классифицирует типы траекторий (прогрессивная, стабильная, волнообразная, регрессивная), проводит кластеризацию учеников, визуализирует прогресс с

помощью Gantt-диаграмм и хуChart, а также прогнозирует риски стагнации, регресса или закрепления ошибок.

Генерация дифференцированных корректирующих заданий осуществляется посредством умения `didactic-task-generator`. На основе результатов анализа агент создает учебные задачи различного уровня сложности (базовый, повышенный, высокий), ориентируясь на таксономию учебных целей Блума-Андерсона и структурно-ментальные схемы Пака и Асауленко.

+ Thought: 316ms

Сформирован файл `tests/didactic-task-generator/result_tasks_for_students.json`.

Всего сгенерировано 12 заданий для 5 учеников по результатам диагностики:

Ученик	Уровень	Задач	Что дано
Студент 2	фактологический (частично)	3	Тест на определение, сопоставление 6 свойств, анализ ситуаций на нарушение свойств
Студент 5	фактологический (частично)	3	Выбор алгоритма из текстов, алгоритм «Собраться в школу» (бытовой), тест на базовые понятия
Студент 1	концептуальный	3	Заполнение таблицы 6 свойств, задача на вспомогательный алгоритм (формула Герона), структурно-ментальная схема
Студент 3	процедурный	3	Блок-схема квадратного уравнения (ветвление), алгоритм с циклом <code>while</code> (среднее арифметическое), задача на СКИ исполнителя «Робот»
Студент 4	творческий	3	Анализ эффективности (рекурсия vs итерация, 0-нотация), машина Тьюринга для сложения, алгоритмическая неразрешимость

■ Build · Big Pickle · 2m 2s

Рисунок 4 – Пример работы умения `didactic-task-generator`.

Задание 3 — Уровень: БАЗОВЫЙ | Тип: тест | Время: 5 мин | Max балл: 4**Условие:**

Закончите предложения, выбрав правильный вариант:

1. Алгоритм — это:
 - а) последовательность команд исполнителю
 - б) любой текст
 - в) математическая формула

2. Исполнитель алгоритма — это:
 - а) тот, кто составляет алгоритм
 - б) тот, кто выполняет команды алгоритма
 - в) компьютерная программа

3. Команда алгоритма — это:
 - а) любое предложение
 - б) одно действие, которое может выполнить исполнитель
 - в) результат алгоритма

4. Результат алгоритма — это:
 - а) то, что получается после выполнения всех команд
 - б) первая команда
 - в) исходные данные

Рисунок 5 – Пример сгенерированного комплекта заданий с помощью умения didactic-task-generator.

Благодаря интеграции указанных умений агент классифицирует пробелы по категориям: отсутствие или искажение ключевых атрибутов, разрыв функциональных связей, недостаточная операционализация понятия, слабая генерализация или неумение применять ментальные схемы в новых ситуациях. В результате формулируются педагогически корректные, понятные учителю выводы, ориентированные на зону ближайшего развития ученика. Например, при анализе понятия «алгоритм» агент способен указать, что ученик успешно усвоил свойства дискретности, определенности и результативности, однако не овладел понятием «ветвление» как необходимой частью-целым структуры алгоритма, что препятствует решению задач с

условным выполнением команд. Такой подход трансформирует диагностику в инструмент развития, полностью созвучный идеям А.В. Усовой о диалектическом единстве объема и содержания понятия и ментальной таксономии Н.И. Пака.

Модель представления комплексного понятия внутри системы задается как многоуровневая структура, включающая операциональное определение в рамках школьного курса информатики, ключевые атрибуты и свойства, иерархию связанных понятий с типами связей, типичные примеры и контрпримеры, а также алгоритмы перехода между уровнями таксономии учебных целей (помнить, понимать, применять, анализировать, оценивать, создавать). Учитель обладает возможностью настраивать или уточнять эту модель под конкретную тему, класс, региональные особенности образовательной программы или индивидуальные методические предпочтения, что обеспечивает высокую гибкость и адаптивность инструмента. Такая модель опирается на синтез фундаментальных идей А.В. Усовой о диалектике общего и единичного, конкретного и абстрактного с современными когнитивными инструментами, обоснованными в главе 1, включая структурно-ментальные схемы для расчетных, операционных и алгоритмических задач. В результате агент не только диагностирует текущее состояние, но и прогнозирует потенциальные затруднения.

Сценарий работы учителя с агентом представляет собой последовательный, строго алгоритмизированный цикл, полностью соответствующий принципам системно-деятельностного подхода.

На первом шаге педагог задает тему, класс и целевое понятие, вводит или загружает ответы учащихся (индивидуально или по группе). Агент выполняет декомпозицию, семантическое сопоставление и многоуровневый анализ, формируя структурированный отчет с визуализацией пробелов. На основе полученных данных учитель получает рекомендации по коррекции и при необходимости запускает генерацию дополнительных заданий от простых расчетных задач на базовом уровне до творческих алгоритмических на продвинутом. Завершающий этап включает профессиональную рефлекссию педагога, фиксацию результатов для дальнейшего мониторинга динамики формирования понятий и корректировку индивидуальных образовательных траекторий. Весь процесс укладывается в логику формирующего оценивания, минимизирует временные затраты и повышает эффективность педагогического труда в условиях ограниченного количества учебных часов.

Вместе с тем, разработанная система обладает четко очерченными ограничениями и рамками использования, что принципиально важно для сохранения гуманистической направленности и этической корректности педагогического процесса. ИИ-агент выступает исключительно в роли помощника учителя, а не его замены; окончательная интерпретация результатов, принятие педагогических решений, учет индивидуальных

психологических особенностей учащихся и этическая оценка остаются безусловной прерогативой профессионального педагога. Агент не претендует на абсолютную точность в случаях неоднозначных, креативных, контекстно-зависимых или эмоционально окрашенных ответов учащихся, особенно на продвинутых уровнях формирования понятий. Результативность работы агента напрямую обусловлена качеством исходных данных, полнотой и своевременностью обновления эталонных моделей, а также уровнем педагогической рефлексии и цифровой грамотностью учителя. Вместе с тем система преимущественно нацелена на диагностику в рамках школьной информатики и без активного вмешательства преподавателя и дополнительного инструментария не охватывает эмоциональное состояние, мотивационную сферу, социальные или метапредметные навыки. Строгое следование принципам объяснимости, прозрачности и «учитель в контуре управления» (teacher-in-the-loop), обоснованным в разделе 1.3, гарантирует, что педагог сохраняет полный методологический и этический контроль над агентом, что сводит к минимуму угрозу алгоритмической предвзятости и необоснованного доверия к автоматическим рекомендациям.

Следовательно, разработанный комплекс навыков ИИ-агента представляет собой практическую реализацию теоретических положений первой главы, превращая формализованные подходы к алгоритмизации диагностической деятельности учителя в действующее технологическое решение. Она обеспечивает переход от фрагментарного контроля к

системному анализу структуры и динамики понятийных образований, сохраняя при этом ведущую роль педагога в интерпретации результатов и принятии педагогических решений. Полученные разработки закладывают надежную основу для экспериментальной апробации агента и дальнейшего совершенствования его функциональности в контексте подготовки будущих учителей информатики.

2.2. Установка и запуск ИИ-агента

В развитие положений предыдущих параграфов второй главы, посвящённых разработке системы команд и навыков ИИ-агента, возникает необходимость в строгой формализации процедур внедрения программного комплекса в реальную педагогическую среду. Для учителя информатики, выступающего основным пользователем и часто не обладающего глубокими навыками системного администрирования, наличие чёткого, пошагового операционального алгоритма приобретает принципиальное значение. Настоящий параграф раскрывает технические аспекты установки, развёртывания и первичной настройки ИИ-агента на базе OpenCode максимально доступным образом. Полный архив агента со всеми навыками, конфигурационными файлами и структурой проекта доступен по ссылке на Яндекс Диск: [37]. Алгоритм ориентирован на занятого педагога и обеспечивает формирование процедурного, метакогнитивного и творческого уровней владения инструментом в соответствии с деятельностным подходом А. Н. Леонтьева [12].

Установка начинается с подготовки операционной системы и проверки аппаратных требований. Затем устанавливаются среды программирования python и node.js. Далее устанавливается агент OpenCode средствами node.js и выполняется его тестовый запуск. После этого скачиваются и устанавливаются проектные зависимости, а затем – разработанный авторский ИИ-агент.

Установка необходимых компонентов. Вначале необходимо установить Node.js с официального сайта <https://nodejs.org/>, обязательно отметив при установке опцию «Add to PATH» (Рисунок 6), и Python (Рисунок 7) с официального сайта <https://www.python.org/downloads/>, также с обязательным добавлением в PATH.

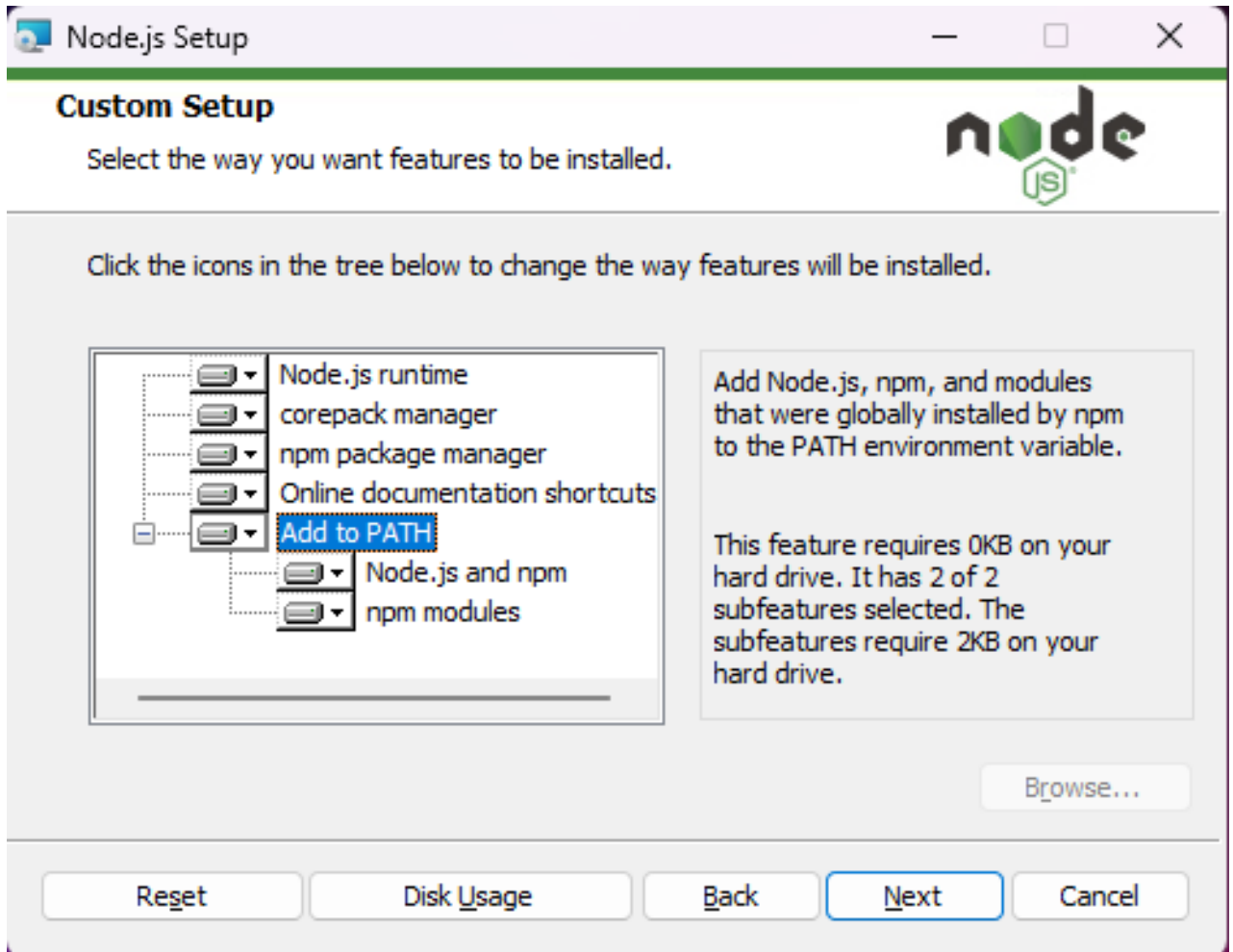


Рисунок 6 – Окно установки Node.js



Рисунок 7 – окно установки Python

После установки рекомендуется открыть окно PowerShell от имени администратора и проверить версии компонентов командами `node --version`, `npm --version` и `python --version` (Рисунок 8). Желательно также установить Graphviz [28] для корректной генерации семантических графов понятий.

```

C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.26200.8457]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\kirit>node --version
v24.16.0

C:\Users\kirit>npm --version
11.12.1

C:\Users\kirit>python --version
Python 3.14.4

C:\Users\kirit>

```

Рисунок 8 – Проверка версий Node.js, Python и NMP

Если при установке Node.js или Python опция «Add to PATH» не была отмечена, пути добавляются вручную. Для этого надо открыть «Настройки» → «Свойства» → «Дополнительные параметры системы» → «Переменные среды».

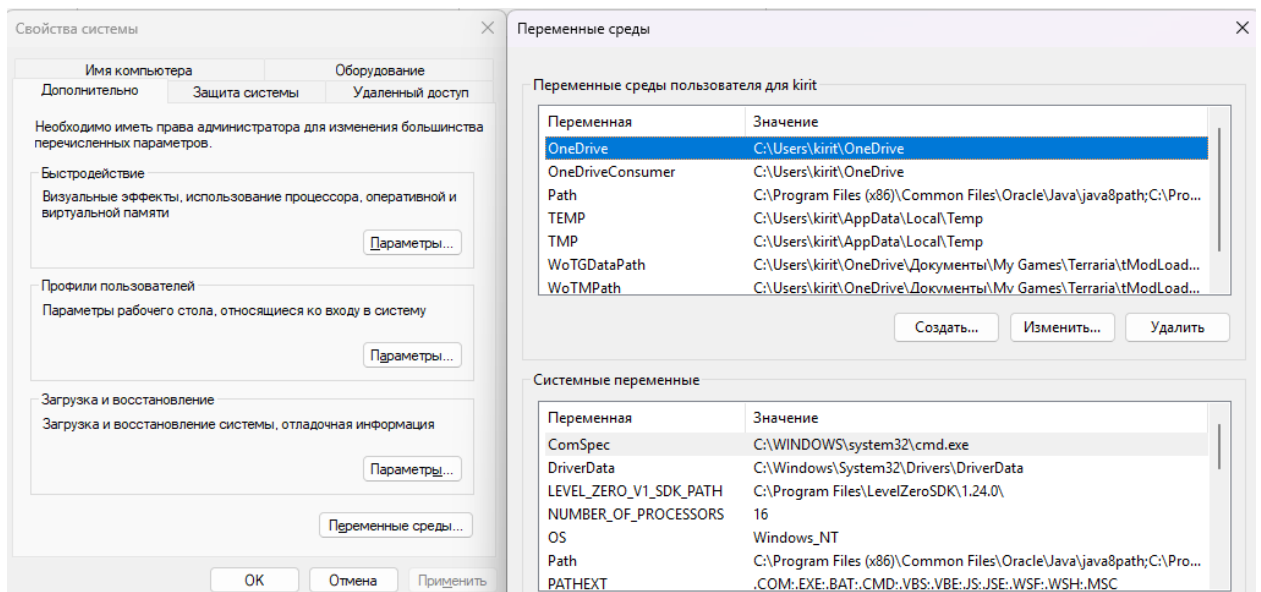


Рисунок 9 – Переменные среды

В разделе «Системные переменные» находят переменную Path, нажимают «Изменить» и добавляют следующие строки: для Node.js *C:\Program Files\nodejs*; для Python *C:\Users\ИмяПользователя\AppData\Local\Programs\Python\Python314* (или аналогичный путь к папке с python.exe). После внесения изменений необходимо перезапустить PowerShell.

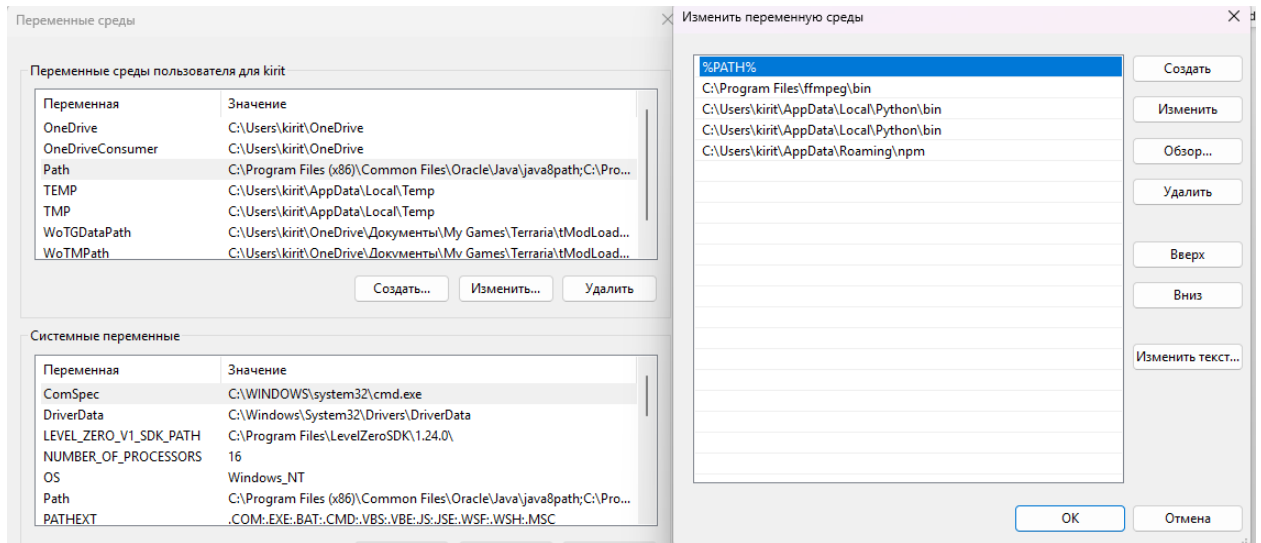


Рисунок 10 – Добавление Node.js и Python в Path

Затем устанавливаются проектные зависимости: `npm install --save-dev @mermaid-js/mermaid-cli@^11.15.0`.

После установки необходимых компонентов для работы агента, выполняется глобальная установка CLI-утилиты в окне PowerShell, запущенном от имени администратора, командой `npm install -g @opencode-ai/cli` (Рисунок 11), после чего рекомендуется выполнить `opencode init` (если структура не была создана автоматически).

```

npm install @opencode-ai/cli
Windows PowerShell
(C) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

Установите последнюю версию PowerShell для новых функций и улучшения! https://aka.ms/PSWindows

PS E:\opencode ai\opencode_agent> cmd
Microsoft Windows [Version 10.0.26200.8457]
(C) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

E:\opencode ai\opencode_agent>npm install -g @opencode-ai/cli
||

```

Рисунок 11 – Установка агента

Далее следует скачать архив проекта `opencode_agent.zip` по предоставленной ссылке на Яндекс Диск [37] и разархивировать его в удобную папку, например `C:\opencode_agent` (Рисунок 12).

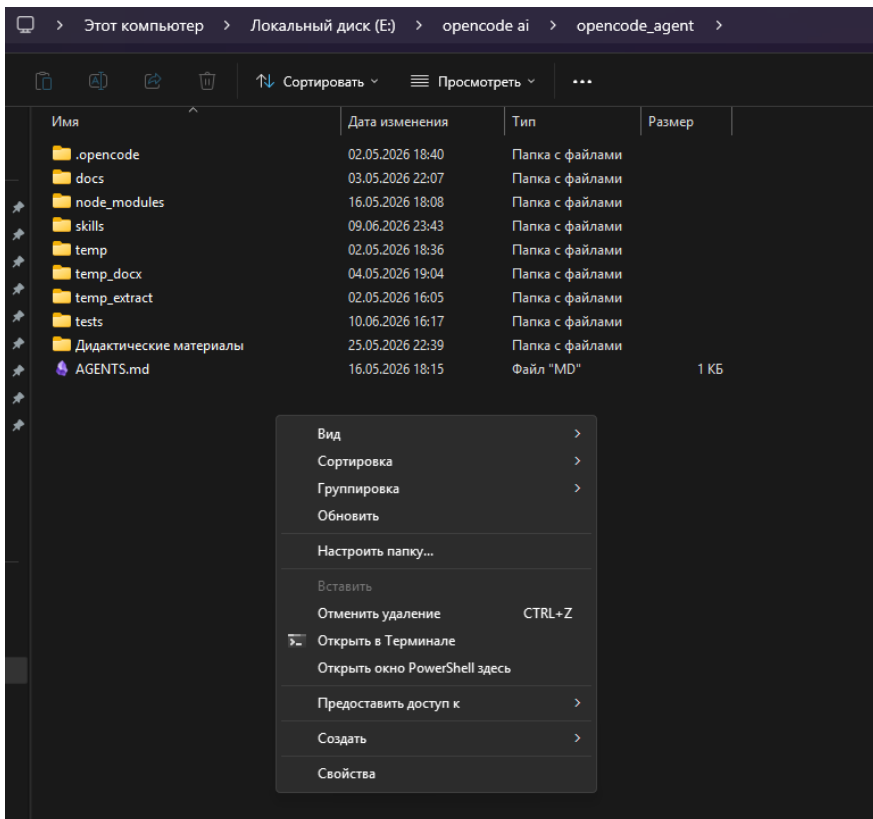
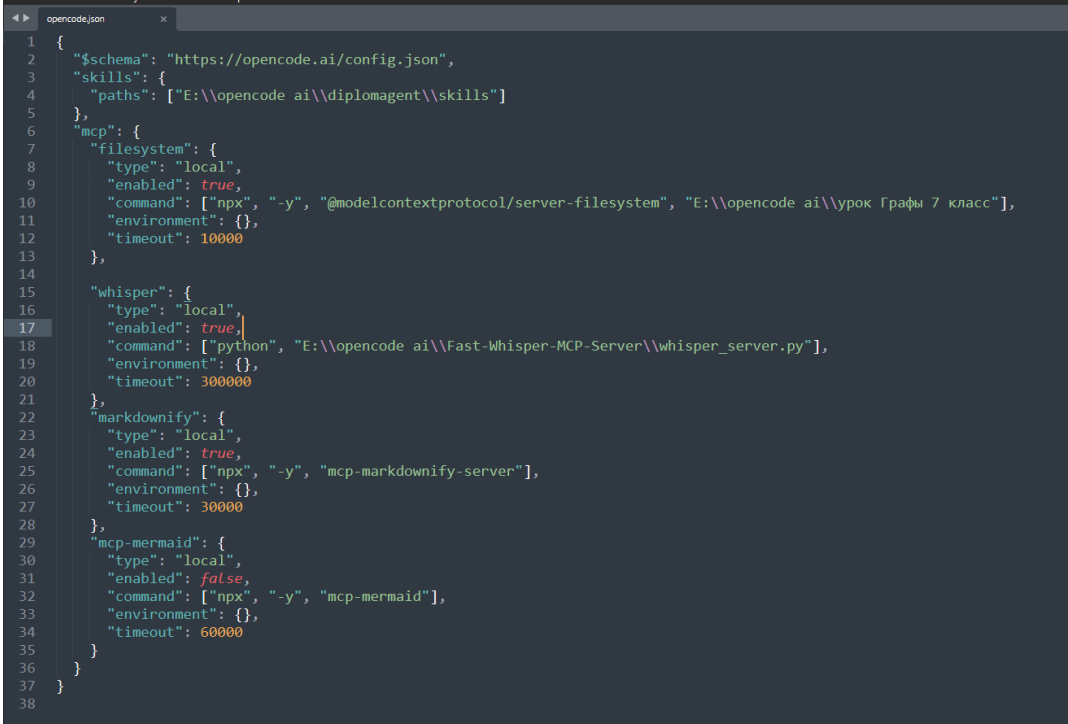


Рисунок 12 – Папка структуры агента после распаковки

Первичная конфигурация осуществляется в файле `C:\opencode_agent\opencode\opencode.json` (Рисунок 13), где уточняются пути к папкам проекта и команды запуска MCP-серверов (`filesystem`, `whisper`, `markdownify`). В частности, путь к файловому серверу должен указывать на `C:\opencode_agent`. Формируется профиль пользователя с указанием предметной области «информатика», целевых классов и предпочтительных форматов отчётов. Права доступа настраиваются через свойства папок `C:\opencode_agent\skills\` (Рисунок 14), `C:\opencode_agent\tracker-data\` и `C:\opencode_agent\Дидактические материалы\`.



```
1 {
2   "$schema": "https://opencode.ai/config.json",
3   "skills": {
4     "paths": ["E:\\opencode ai\\diplomagent\\skills"]
5   },
6   "mcp": {
7     "filesystem": {
8       "type": "local",
9       "enabled": true,
10      "command": ["npx", "-y", "@modelcontextprotocol/server-file-system", "E:\\opencode ai\\урок Графы 7 класс"],
11      "environment": {},
12      "timeout": 10000
13    },
14  },
15  "whisper": {
16    "type": "local",
17    "enabled": true,
18    "command": ["python", "E:\\opencode ai\\Fast-Whisper-MCP-Server\\whisper_server.py"],
19    "environment": {},
20    "timeout": 300000
21  },
22  "markdownify": {
23    "type": "local",
24    "enabled": true,
25    "command": ["npx", "-y", "mcp-markdownify-server"],
26    "environment": {},
27    "timeout": 30000
28  },
29  "mcp-mermaid": {
30    "type": "local",
31    "enabled": false,
32    "command": ["npx", "-y", "mcp-mermaid"],
33    "environment": {},
34    "timeout": 60000
35  }
36 }
37 }
38 }
```

Рисунок 13 – Структура файла `opencode.json`

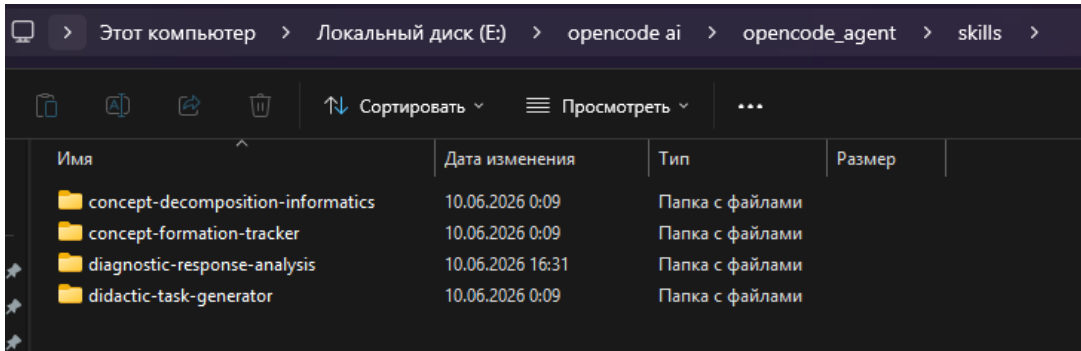


Рисунок 14 – Структуры папки skills

Настройка предметного контекста предполагает создание (при необходимости) папки *C:\oencode_agent\Дидактические материалы* (Рисунок 15), куда копируются ФГОС, рабочие программы, учебники и методические пособия. Это позволяет навыкам, в частности *concept-decomposition-informatics*, опираться на достоверной источники при генерации семантических графов понятий.

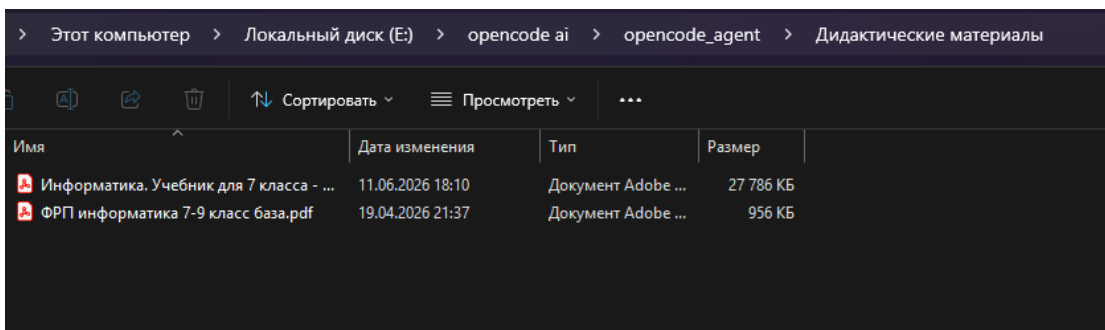


Рисунок 15 – Папка «Дидактические материалы»

Проверка работоспособности выполняется непосредственно в папке *C:\oencode_agent* командой *oencode* в командной строке. Успешный тестовый прогон включает генерацию графа понятия и анализ демонстрационного ответа учащегося, что подтверждается отсутствием критических ошибок и корректностью формируемых JSON-структур.



Рисунок 16 – Запуск агента

Рекомендуется регулярное резервное копирование всей папки C:\opencode_agent и использование BitLocker для дополнительной защиты.

Установка завершена.

В качестве примера рассмотрим ситуацию, при которой преподавателю необходимо визуализировать дерево понятий для темы «Файловая система». Для этого в директорию «Дидактические материалы» помещаются соответствующие источники: учебник для нужного уровня обучения (например, Босова, 7–9 классы), федеральная рабочая программа, а также, при необходимости, кодификаторы ОГЭ/ЕГЭ, позволяющие учесть перечень проверяемых на экзамене понятий.

После подготовки материалов преподавателю следует запустить агента и сформировать запрос в интерфейсе системы, например: «Разбери понятие “Файловая система” и построй граф» (Рисунок 17).

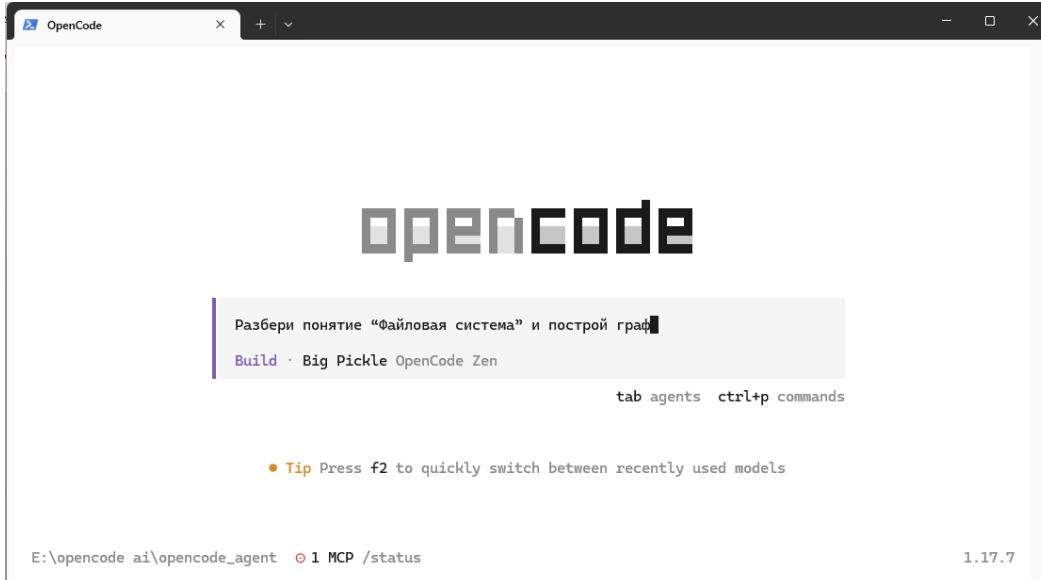


Рисунок 17 – Пример промта учителя агенту

При этом запросе агент сам вызывает умение `concept-decomposition-informatics`. После чего учитель получает готовый граф понятия «Файловая система» изображением в формате PNG (Рисунок 18).

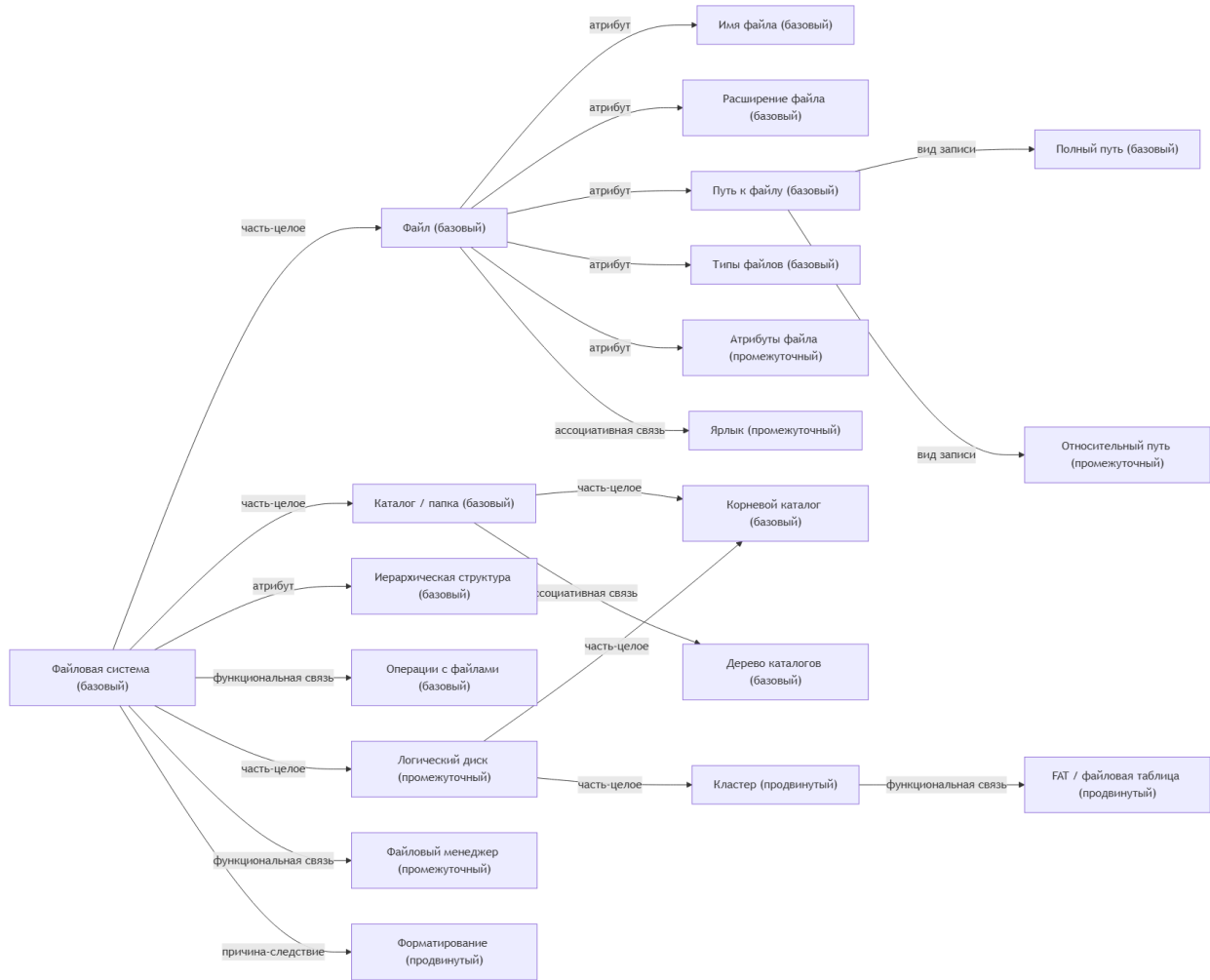


Рисунок 18 – Готовый граф для понятия «Файловая система»

Заключение

В выпускной квалификационной работе выполнено теоретико-методологическое обоснование и практическая разработка специализированного ИИ-агента, предназначенного для автоматизации анализа сформированности комплексных понятий школьного курса информатики. Исследование направлено на решение ключевой задачи современного педагогического образования – автоматизации диагностической деятельности учителя в условиях цифровой трансформации, где традиционные формы контроля знаний демонстрируют ограниченную эффективность при работе со сложными, системными понятийными структурами.

В первой главе раскрыты теоретические основы применения ИИ-агентов в педагогических процессах информационной обработки. Опираясь на работы А. Н. Леонтьева и Л. С. Выготского, показано, что ИИ-агент следует внедрять не как средство простой автоматизации, а как когнитивного посредника, расширяющий операциональную сферу педагога и обеспечивающий динамический мониторинг формирования понятий в зоне ближайшего развития обучающихся. Процесс педагогической деятельности учителя алгоритмизирован, а ключевые действия, выполняемые педагогом при диагностике комплексных понятий, формализованы и представлены в виде умений агента. Данный подход направлен на практическую реализацию ментального подхода Н. И. Пака, поскольку соответствующие умения обеспечивают построение эталонного семантического графа и графа знаний

обучающегося, на основе сопоставления которых формируются выводы о состоянии и структуре его ментальной модели предметной области. В соответствии с полученными результатами педагог получает возможность осуществлять дальнейшую корректировку образовательного процесса.

Сам агент, процесс его установки и процесс его использования для решения педагогических задач учителя описаны в соответствующих разделах главы 2. Ключевые умения следующие:

Таблица 2

Ключевые умения агента

Умение (Skill)	Используется для
concept-decomposition-informatics	декомпозиций комплексных понятий с построением и визуализацией семантического графа
diagnostic-response-analysis	многоуровневого анализа ответов по таксономии ментального подхода
concept-formation-tracker	систематического отслеживания динамики формирования понятий
didactic-task-generator	создания дифференцированных дидактических заданий

Сформулированы подробные рекомендации по внедрению ИИ-ассистента в практику учителя информатики. Рекомендации охватывают алгоритм установки и первичной настройки в среде Windows, конфигурацию МСР-серверов, обеспечение информационной безопасности в соответствии с

Федеральный закон № 152-ФЗ (в части защиты персональных данных) соблюдается при использовании локальных LLM-модели, поскольку архитектура таких моделей изначально по дизайну предполагает обработку данных в локальном контуре без их передачи в внешние сервисы, что реализуемо в решениях с открытым исходным кодом.

Есть ряд перспективных направлений для дальнейшей работы. К ним относятся расширение функциональности агента на метапредметные результаты, апробация в условиях реального обучения, разработка механизмов адаптивного дообучения на данных конкретных образовательных организаций, а также углублённое изучение влияния длительного использования ИИ-ассистента на профессиональную рефлексию и мышление учителей.

Таким образом, в работе реализован целостный цикл от теоретического обоснования до практической разработки и методического сопровождения ИИ-агента для анализа сформированности комплексных понятий. Созданный инструмент выступает не только технологическим решением, но и педагогическим средством, усиливающим развивающую функцию образования. Полученные результаты вносят вклад в решение задач цифровой трансформации российского образования и создают научно-методическую основу для дальнейшего совершенствования систем поддержки педагогической диагностики в условиях информационного общества.

Список использованных источников

1. Асауленко Е. В., Баженова И. В., Клунникова М. М., Пак Н. И. Структурно-ментальные схемы как когнитивные инструменты формирования и развития учебных компетенций // Педагогическая информатика. 2020. № 3. С. 3–15.
2. Баранова Е. В., Симонова И. В. Система учебных задач для формирования алгоритмической компетенции при подготовке учителей информатики // Вестник Герценовского университета. 2020. № 2. С. 44–53.
3. Босова Л. Л. и др. Актуальные проблемы методики обучения информатике и математике в современной школе. //М: МПГУ, 2020. 320 с.
4. Ватьян А. С., Гусарова Н. Ф., Добренко Н. В. Системы искусственного интеллекта. //СПб. : Университет ИТМО, 2022. 312 с.
5. Выготский Л. С. Мышление и речь. //Москва : Лабиринт, 1999. 352 с.
6. Заславская О. Ю., Галеева Н. Л. Подходы к управлению учебной деятельностью учащихся на уроках информатики // Информатика и образование. 2020. № 6. С. 30–38.
7. Камалова Г. Б. Вычислительная информатика в системе подготовки будущих учителей информатики // Педагогическая информатика. 2021. № 2. С. 45–53.
8. Клепшман М. Проектирование систем обработки данных. //СПб.: Питер, 2018.
9. Левченко И. В. Методика обучения информатике на уровне основного общего образования. //Москва : МПГУ, 2021. 248 с.
10. Левченко И. В., Садыкова А. Р., Абушкин Д. Б., Карташова Л. И., Кондратьева В. А., Моисеев В. П. Особенности подготовки по программированию будущих учителей информатики // Информатика и образование. 2022. № 4. С. 14–26.
11. Левченко И. В., Садыкова А. Р., Карташова Л. И., Меренкова П. А. Методологические и методические основы разработки технологических

- карт уроков информатики // Педагогическое образование в России. 2021. № 3. С. 58–67.
12. Леонтьев, А. Н. Деятельность. Сознание. Личность / А. Н. Леонтьев. // М. Политиздат, 1975. — 304 с.
13. Намиот Д. Е., Ильюшин Е. А. О кибербезопасности ИИ-агентов // International Journal of Open Information Technologies. 2025. Т. 13, № 9. С. 45–62.
14. Нефедова В. Ю., Киякбаева К. К. Методические особенности обучения алгоритмизации и программированию в школьном курсе информатики // Информатика в школе. 2021. № 5. С. 12–19.
15. Пак Н. И. Таксономия учебных целей с позиций ментального подхода // Педагогическая информатика. 2019. № 1. С. 21–31.
16. Панкратов И. Ю. Использование ИИ-агентов в управлении научно-исследовательской деятельностью на примере онлайн-платформы «Киберпрофессор» // Государственная служба. 2025. № 2. С. 34–41.
17. Руководство по использованию генеративного искусственного интеллекта в образовании и научных исследованиях. // Париж : ЮНЕСКО, 2024. 56 с.
18. Сардалова Л. Р. Инновационные методологии в образовательной сфере на примере внедрения искусственного интеллекта в систему высшего образования // Вестник научных конференций. 2024. № 3-2 (93). С. 78–82.
19. Титова С. В. Интеллектуальные агенты в обучении ИЯ: типология, возможности, вызовы // Педагогические науки. 2024. № 4. С. 112–119.
20. Усова А. В. Теоретические основы формирования научных понятий в процессе обучения. // Челябинск : ЧГПУ, 1997. 100 с.
21. About Mermaid [Электронный ресурс]. – URL: <https://mermaid.js.org/intro/> (дата обращения: 17.06.2026).
22. Arike R., Donoway E., Bartsch H., Hobbhahn M. Technical Report: Evaluating Goal Drift in Language Model Agents. 2025.
23. AutoGPT Documentation [Электронный ресурс]. – URL: <https://agpt.co/docs> (дата обращения: 14.05.2026).

24. Boston Consulting Group (BCG). How AI Agents and Tech Will Transform Health Care in 2026. //Boston: BCG, 2025.
25. Cervantes H., Kazman R. Designing Software Architectures: A Practical Approach, 2nd Edition. //Boston: Addison-Wesley Professional, 2024.
26. CrewAI Documentation [Электронный ресурс]. – URL: <https://docs.crewai.com/> (дата обращения: 13.05.2026).
27. Dong Y., Mu R., Jin G., Qi Y., Hu J., Zhao X., Meng J., Ruan W., Huang X. Building Guardrails for Large Language Models. 2024.
28. Graphviz Documentation [Электронный ресурс]. – URL: <https://graphviz.org/documentation/> (дата обращения: 17.06.2026).
29. Jestus Sahaya, Aryal Lazer. A Survey of Agentic AI and Cybersecurity: Challenges, Opportunities and Use-case Prototypes. 2025.
30. Kaptein M., Khan V.-J., Podstavnychy A. Runtime Governance for AI Agents: Policies on Paths. 2026.
31. Khamis A. Agentic AI Systems: Architecture and Evaluation Using a Frictionless Parking Scenario // IEEE Access. 2025
32. Lewis P., Perez E., Piktus A., Petroni F., Karpukhin V., Goyal N., Kuttler H., Lewis M., Yih W.-t., Rocktäschel T., Riedel S., Zettlemoyer W. Retrieval-Augmented Generation for Knowledge-Driven NLP. 2020.
33. Liang, G., & Tong, Q. LLM-Powered AI Agent Systems and Their Applications in Industry. 2025.
34. Model Context Protocol (MCP) [Электронный ресурс] / Anthropic. – 2024. – URL: <https://modelcontextprotocol.io> (дата обращения: 17.06.2026).
35. NIST AI 100-1 Artificial Intelligence Risk Management Framework (AI RMF 1.0). //Gaithersburg: National Institute of Standards and Technology, 2023. 78 с.
36. OpenCode AI Documentation: Models, Providers, and Configuration [Электронный ресурс]. – URL: <https://opencode.ai/docs/ru/models/> (дата обращения: 10.05.2026).

37. `Opencode_agent.zip` [Электронный ресурс]. – URL: <https://disk.yandex.ru/d/5T8GYvLb7xdz1g> (дата обращения: 15.06.2026)
38. Wang C. L., Singhal T., Kelkar A., Tuo J. MI9: An Integrated Runtime Governance Framework for Agentic AI. 2025.
39. Wei J., Wang X., Schuurmans D., Bosma M., Chi E.H., Le Q.V., Zhou D. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. 2022.
40. Xue E., Chen K., Huang Z., Ji Y., Wang H. IMPROVE: Iterative Model Pipeline Refinement and Optimization Leveraging LLM Agents. arXiv:2502.18530, 2025.
41. Yao S., Zhao J., Yu M., Niekumen S., Klein D., Binder E., Lewitt M. ReAct: Synergizing Reasoning and Acting in Language Models. 2023.

Приложение
к Регламенту размещения
выпускной квалификационной работы обучающихся,
по основным профессиональным образовательным программам
в КГПУ им. В.П. Астафьева

Согласие
На размещение выпускной квалификационной работы обучающегося
в ЭБС КГПУ им. В.П. Астафьева

Я, Голубков Константин Алексеевич

(фамилия, имя, отчество)

разрешаю КГПУ им. В.П. Астафьева безвозмездно воспроизводить и размещать (доводить до всеобщего сведения) в полном объёме и по частям написанную мною в рамках выполнения основной профессиональной образовательной программы выпускную квалификационную работу бакалавра / специалиста / магистра / аспиранта

(нужное подчеркнуть)

на тему: Агент для тестирования уровня сформированности комплексных понятий школьного курса информатики


(название работы)

(далее – ВКР) в сети Интернет в ЭБС КГПУ им. В.П. Астафьева, расположенном по адресу <http://elib.kspu.ru>, таким образом, чтобы любое лицо могло получить доступ к ВКР из любого места и в любое время по собственному выбору, в течение всего срока действия исключительного права на ВКР.

Я подтверждаю, что ВКР написана мною лично, в соответствии с правилами академической этики и не нарушает интеллектуальных иных лиц.

22.05.2026

(дата)



(подпись)

Отзыв научного руководителя Романова Дмитрия Валерьевича на выпускную квалификационную работу студента 5 курса ИМФИ КГПУ им. В.П. Астафьева Голубкова Константина Алексеевича «Агент для тестирования уровня сформированности комплексных понятий школьного курса информатики»

Работа Константина Алексеевича выполнена как часть общего с Елгиной Анастасией Игоревной проекта, и адресует проблему трансформации работы педагога, вызванную широким распространением ИИ-технологий среди работников умственного труда. Искусственный интеллект (ИИ) впервые предоставил возможность перевести целый ряд теоретических наработок из качественных в количественные и практически применимые.

Анастасия Игоревна определила и проработала понятие "комплексное понятие (школьного курса информатики)". Константин Алексеевич перенёс разработки Пака Н.И., Асауленко Е.В. и др. в практическую плоскость, создав ИИ-агента, способного визуализировать ментальную схему по любому набору источников, от ФГОС до контрольной работы ученика. Это позволит педагогу лучше анализировать ход учебного процесса, проектировать занятия и принимать решения. Соискатель определил место применения технологии в учебном процессе и выполнил все поставленные задачи. Так же Константин Алексеевич творчески переработал большой массив печатных работ как по методам проектирования и использования ИИ-агентов, так и по педагогике.

Сам Константин Алексеевич проявил себя как находчивый, самостоятельный, инициативный и работоспособный специалист, продемонстрировал высокий уровень подготовки как педагога, умение развиваться, успешно освоил и применил в ходе работы целый ряд новых технологий.

Созданный инструмент будет использоваться в ИМФИ и «Технопарке универсальных педагогических компетенций», имеет теоретическую и практическую значимость.

Считаю, что работа удовлетворяет всем требованиям, предъявляемым к выпускным квалификационным работам в ИМФИ КГПУ им. В.П. Астафьева, и заслуживает оценки «отлично», а её автор, Голубков Константин Алексеевич, достоин присвоения квалификации бакалавр по направлению подготовки 44.03.05 «Педагогическое образование» с двумя профилями подготовки («Математика» и «Информатика»).

Канд. физ.-мат. наук., доцент кафедры ИИТО



Романов Д. В.

СПРАВКА

о результатах проверки текстового документа
на наличие заимствований

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ
БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
"КРАСНОЯРСКИЙ ГОСУДАРСТВЕННЫЙ
ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ ИМ. В.П.
АСТАФЬЕВА"

ПРОВЕРКА ВЫПОЛНЕНА В СИСТЕМЕ АНТИПЛАГИАТ.ВУЗ

Автор работы: Голубков Константин Алексеевич
Самоцитирование
рассчитано для: Голубков Константин Алексеевич
Название работы: Агент для тестирования уровня сформированности комплексных понятий школьного курса информатики
Тип работы: Выпускная квалификационная работа
Подразделение:

РЕЗУЛЬТАТЫ

- **ВНИМАНИЕ, ДОКУМЕНТ ПОДОЗРИТЕЛЬНЫЙ:** ОБНАРУЖЕНЫ ПОПЫТКИ МАСКИРОВКИ ЗАИМСТВОВАНИЙ. РЕКОМЕНДУЕТСЯ ПРОВЕРИТЬ ПОЛНЫЙ ОТЧЕТ
- **ОТЧЕТ О ПРОВЕРКЕ КОРРЕКТИРОВАЛСЯ:** НИЖЕ ПРЕДСТАВЛЕНЫ РЕЗУЛЬТАТЫ ПРОВЕРКИ ДО КОРРЕКТИРОВКИ

СОВПАДЕНИЯ	0%	СОВПАДЕНИЯ	0.75%
ОРИГИНАЛЬНОСТЬ	1.52%	ОРИГИНАЛЬНОСТЬ	99.25%
ЦИТИРОВАНИЯ	0%	ЦИТИРОВАНИЯ	0%
САМОЦИТИРОВАНИЯ	98.48%	САМОЦИТИРОВАНИЯ	0%
ИИ-КОНТЕНТ	9.2%		

ДАТА И ВРЕМЯ КОРРЕКТИРОВКИ: 19.06.2026 06:38

ДАТА ПОСЛЕДНЕЙ ПРОВЕРКИ: 18.06.2026

Структура документа:

Проверенные разделы: основная часть с.39-57, 66-89, введение с.3-38, выводы с.58-66

Модули поиска:

Переводные заимствования; Профессиональная лексика. Медицина; СМИ России и СНГ; Коллекция НБУ; Перефразированные заимствования по коллекции Интернет в английском сегменте; Патенты СССР, РФ, СНГ; Профессиональная лексика. АПК и биотех; ИПС Адилет; Сводная коллекция ЭБС; Цитирование; Медицина; Перефразирования по коллекции IEEE; Кольцо вузов; Профессиональная лексика. Юриспруденция; Переводные заимствования IEEE; Шаблонные фразы; Сводная коллекция научных работ Беларуси; Публикации РГБ; Кольцо вузов (переводы и перефразирования); СПС ГАРАНТ: аналитика; IEEE; Публикации РГБ (переводы и перефразирования); Публикации eLIBRARY (переводы и перефразирования); Публикации eLIBRARY; Перефразирования по базе публикаций открытого доступа PubMed; PubMed; Перефразирован...

Заключение о работе (оценка):

Работу проверил: Романов Дмитрий Валерьевич
ФИО проверяющего

Дата подписи:



Подпись проверяющего



Чтобы убедиться
в подлинности справки, используйте QR-код,
который содержит ссылку на отчет.

Ответ на вопрос, является ли обнаруженное заимствование
корректным, система оставляет на усмотрение проверяющего.
Предоставленная информация не подлежит использованию
в коммерческих целях.