

Онтологический подход к проектированию образовательных программ в цифровых средах

Рассмотрены вопросы разработки систем проектирования образовательных программ и их компонентов. Предложен подход к такому проектированию, базирующийся на специальных онтологиях, фиксирующих отношения между компонентами программ в различных парадигмах. Описана возможность инструментальной настройки уровня детализации онтологий. Предложена архитектура системы проектирования учебных планов в рамках построенных онтологий.

Ключевые слова: онтологии в образовании, образовательное проектирование, цифровые образовательные системы

ВВЕДЕНИЕ

Цифровые образовательные среды – это совокупность информационных решений, систем и инструментов, способствующих повышению качества образования, содействующих индивидуализации, персонализации и адаптивности обучения, а также сокращению ручного труда в задачах проектирования и управления образовательной деятельностью.

Образовательное проектирование (*Education Design*) – проработка основных деталей деятельности обучающихся и преподавателей для достижения наилучших образовательных результатов. Под такими результатами понимают компетенции, формируемые в ходе обучения и практики, связанные с ними знания, умения и навыки [1]. Инструменты образовательного проектирования – это неотъемлемая часть цифровых образовательных сред.

Основным результатом образовательного проектирования является *образовательная программа* (далее – ОП). Согласно ФЗ-273 «Об образовании в Российской Федерации»¹ это «комплекс основных характеристик образования (объем, содержание, планируемые результаты)». Таким образом, при образовательном проектировании нужно детально продумывать все составляющие ОП.

Отметим, что разные виды образования содержат различные типы компонентов и требуют своего уровня детализации проектирования. Например, в высшем образовании необходимо брать в расчет компетенции, предусмотренные Федеральными государственными образовательными стандартами (ФГОС) [2], а в корпоративном – трудовые функции и профессиональные

компетенции². Или же при проектировании одной образовательной программы требуется составить только список дисциплин, а в другой – обеспечить детализацию по всем темам всех предметов.

В связи с развитием информационных технологий и цифровизацией образования в последние 20 лет появляются цифровые системы, позволяющие осуществлять образовательное проектирование. К этим системам можно отнести:

1. Системы создания учебных планов и рабочих программ, разрабатываемые вузами [3, 4]. Они позволяют распределять количество часов по дисциплинам и их темам, проектировать виды аудиторной и самостоятельной работ. Реализована проверка соответствия часов с учебным планом. Предоставляется возможность создавать новую рабочую программу или программу на основе уже имеющейся. Некоторые системы [4] позволяют добавлять цели, задачи и требования к результатам освоения дисциплин, а также отмечать несколько дисциплин как связанные.

2. Системы управления обучением (*Learning Management Systems, LMS*), самая известная – *Moodle* [5–7]. Такие системы в общем смысле являются веб-приложениями, с помощью которых можно создавать сайты для онлайн-обучения. Система *Moodle*, например, предоставляет средства для проектирования, создания и управления ресурсами информационно-образовательной среды. Преподаватель самостоятельно, используя лишь подсказки справочной системы, может сформировать электронный курс и в дальнейшем управлять его работой.

¹ Федеральный закон от 29 декабря 2012 г. № 273-ФЗ «Об образовании в Российской Федерации».

² Федеральный закон от 02.05.2015 г. № 122-ФЗ «О внесении изменений в Трудовой кодекс Российской Федерации и статьи 11 и 73 Федерального закона «Об образовании в Российской Федерации».

3. Офисные пакеты для создания документов, таблиц, форм, баз данных, презентаций. Сюда, в первую очередь, могут быть отнесены продукты *Microsoft Office – Word, Excel, Power Point, Access, инструменты Google – Docs, Sheets, Slides* и др..

Указанные инструменты, несмотря на свои возможности, обладают рядом существенных недостатков:

- проектирование осуществляется в виде заполнения форм/таблиц. При этом большое количество связей между компонентами образовательной программы записано только в тексте и не отражается в структуре хранения данных, что затрудняет использование этих связей при совершении запросов к спроектированным ОП;
- не осуществляется проверка корректности/полноты заполнения ОП (включая анализ вводимого текста);
- отсутствует механизм отслеживания повторения одних и тех же образовательных модулей, что может привести к дублированию информации;
- системы проектирования поддерживают только один подход (например, только для университета или только для корпоративного обучения), что может затруднить коммуникацию между представителями бизнеса и образования по вопросам обучения специалистов внутри данных систем;
- традиционное табличное представление учебного плана, образовательной программы, рабочей программы дисциплины не является оптимальным интерфейсом для демонстрации структурных связей между компонентами образовательной программы.

Представленная статья является результатом исследований и разработок, посвященных новому подходу к инструментам создания образовательных программ, учитывающим недостатки существующих систем.

ВЫБОР ПОДХОДОВ К ПРОЕКТИРОВАНИЮ ОБРАЗОВАТЕЛЬНЫХ ПРОГРАММ

Результаты проведенных нами исследований показали необходимость разработки единой цифровой системы проектирования образовательных программ, которая представляет собой приложение, позволяющее пользователю создавать и редактировать компоненты образовательной программы и связи между ними, используя при этом уже имеющиеся образовательные программы как базу знаний.

В качестве главного требования определена возможность проектирования образовательных программ в разных образовательных парадигмах с разной степенью детализации своих компонентов.

Выбраны три подхода, на которых можно, с одной стороны, продемонстрировать различия в компонентах, с другой стороны – на их примере показать возможность реализации всех подходов в рамках одной системы. Поскольку выбранные подходы определяют функциональные возможности проектирования, решено называть их также *режимами работы*.

Подход, основанный на полном соответствии федеральным государственным образовательным стандартам (подход ФГОС). Федеральный государственный образовательный стандарт – совокупность требований, обязательных при реализации основных

ОП высшего профессионального образования образовательными учреждениями, имеющими государственную аккредитацию [2, 8]. ФГОС связан с компетентностной моделью выпускника, которая представляет собой описание набора компетенций, которыми он должен обладать, функций, к выполнению которых он должен быть подготовлен, а также степени готовности к выполнению конкретных обязанностей [9]. ФГОС явным образом указывает на связанность процесса формирования компетенций с образовательными технологиями, а также вводит контроль уровня сформированности компетенций.

Подход «обратного проектирования» (*Backward Design*) [10]. Это метод проектирования ОП, при котором детально прорабатываются образовательные результаты и, исходя из них, выстраиваются применяемые методы и модели обучения.

Подход *Backward Design* тесно связан с задачами развития конкретных компетенций для определенных профессий/специальностей. В этом подходе термин «компетенция» раскрывается, в первую очередь, с точки зрения того, как должен работать человек, обладающий компетенцией, и как следует проявлять ее в конкретных ситуациях. Появляется термин «ситуация», выделенный в отдельный компонент ОП и обозначающий описание профессиональной задачи, адаптированное к образовательному процессу.

Также введен термин «ориентировочная основа действий (ООД)» – описание конкретного алгоритма применения компетенции в заданной роли в определенной ситуации.

Проблемно-ориентированный подход (*Problem-based Learning*) [11], связанный также с концепцией *Instructional design* [12, 13]. Образовательные программы в рамках данного подхода строятся на некотором наборе проблемных задач, которые формируют необходимые компетенции. Ключевым понятием здесь является «проблемная ситуация», создаваемая преподавателем для учебных целей и включающая сложный теоретический или практический вопрос, требующий самостоятельного изучения, расширения знаний, исследований.

Посредством решения проблемной ситуации задачи обучающийся получает необходимые образовательные результаты. При этом дисциплина (преподаваемый предмет) уже не является неотъемлемой частью образовательной программы.

Как можно видеть, подходы, представленные выше, различаются в наборе компонентов образовательной программы и связей между ними, что должно существенно влиять на модель данных, используемую в информационной системе. Механизмы организации данных, учитывающих эту особенность, представлены в следующем разделе.

ФОРМАЛИЗАЦИЯ ПРОЕКТИРОВАНИЯ ОБРАЗОВАТЕЛЬНЫХ ПРОГРАММ С ПОМОЩЬЮ ОНТОЛОГИЙ

Как известно, онтологии – это концептуальные схемы, формализующие некоторую область знаний [14, 15]. В их состав входят классы объектов, их связи и правила (теоремы, ограничения), принятые в этой области.

Для формализации был выбран инструмент онтологий, позволяющий реализовать возможности проектирования образовательных программ в рамках любого подхода, учитывая множество типов / видов связей между компонентами ОП, а также детализировать эти ОП в разной степени. Онтологии были спроектированы для каждого режима работы с учетом степени детализации.

Проанализировав ФГОС, ООП, стандарты и системы создания рабочих программ, мы выявили базовые компоненты классического образовательного процесса:

- компетенции и ЗУН (знания, умения, навыки);
- дисциплины (учебные предметы), темы, дидактические единицы (атомарные составляющие ЗУН, далее – ДЕ);
- формы обучения (лекции, практики, лабораторные и самостоятельные работы);
- формы контроля (промежуточные и итоговые);
- образовательные технологии (формы проведения занятий, непосредственно формирующие компетенции);
- литература и методические материалы, образцы заданий.

На основе анализа связей между сущностями была построена онтология проектирования образовательной программы в режиме работы ФГОС (рис. 1).

Перечень компонентов онтологии при подходе обратного проектирования (Backward Design) имеет пересечение с перечнем для модуля ФГОС. Однако, как было сказано выше, в рамках подхода *Backward Design* компетенции рассматриваются с точки зрения иной модели, для чего вводятся дополнительные сущности:

- ситуация;
- роль – статус обучаемого в конкретной ситуации (разработчик, аналитик, руководитель проекта и т. п.);
- ориентировочная основа действий (ООД) – порядок действий, в рамках указанной ситуации в определенной роли, определяющий успешное применение компетенции.

Отметим, что ООД компетенции в рамках одной и той же ситуации может зависеть от роли. К примеру, в ситуации «Срыв сроков выполнения проекта» для компетенции «Профессиональная коммуникация» ООД в роли «Руководитель проекта» и роли «Разработчик» будут различаться. На рис. 2 представлена онтология для данного подхода.

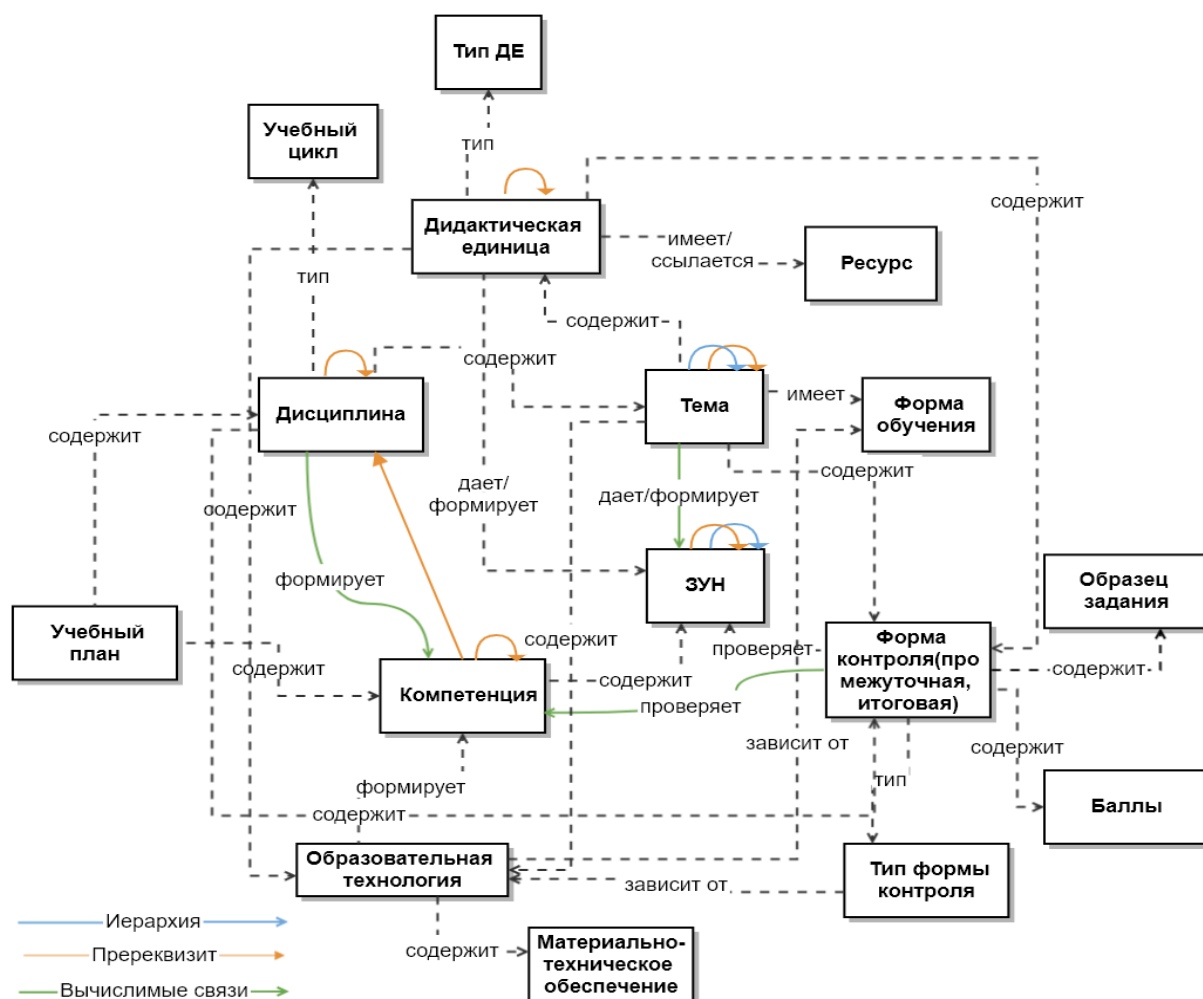


Рис. 1. Построенная онтология системы в рамках подхода, основанного на федеральном государственном образовательном стандарте

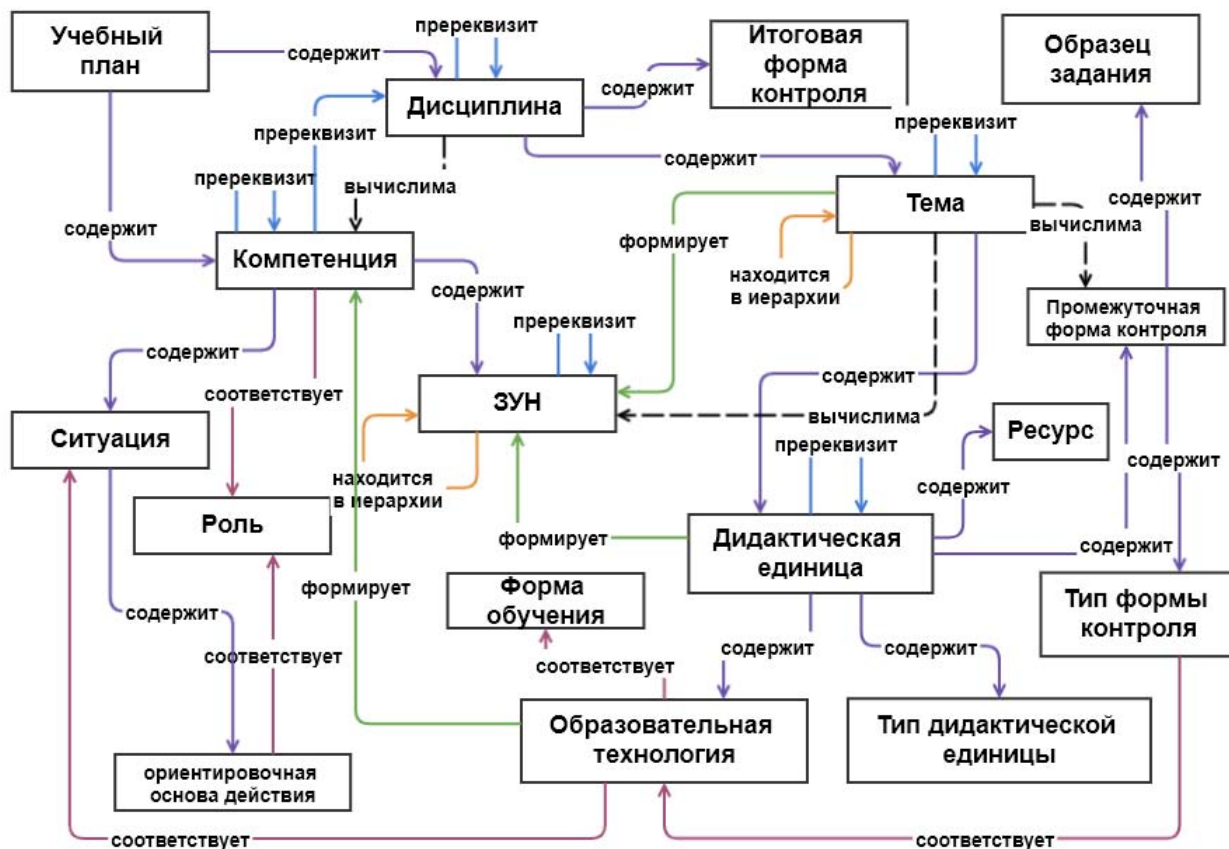


Рис. 2. Построенная онтология системы при подходе *Backward Design*

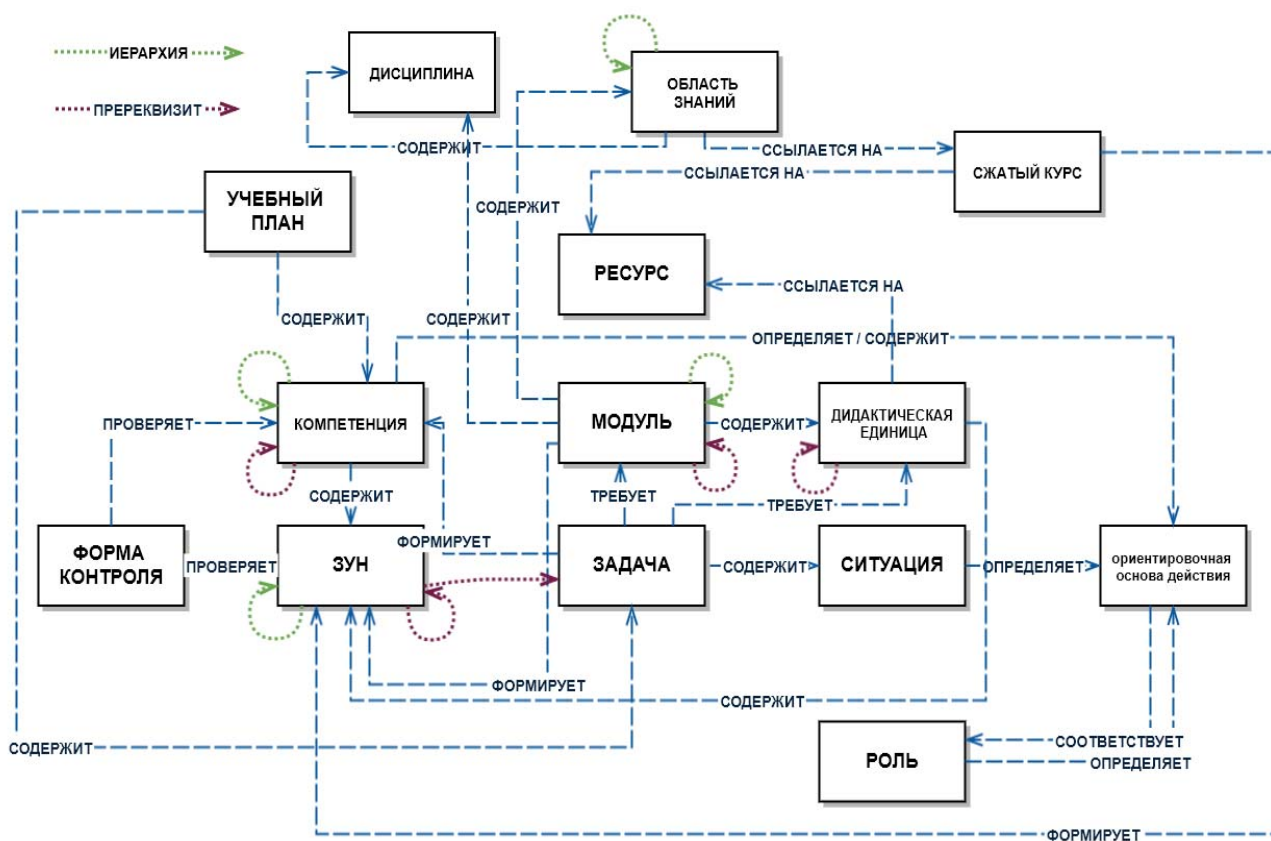


Рис. 3. Построенная онтология системы в рамках подхода *PBL*

При проблемно-ориентируемом подходе (*PBL*) главным инструментом при формировании компетенций служит задача, моделируемая в определенной ситуации. Образование строится из потребности решения проблемных задач, при этом могут изучаться образовательные модули из отдельных дисциплин.

Общая структура онтологии подхода *PBL* (рис. 3) несет следующий смысл – компетенции *приобретаются* в ходе решения задач; задача, в свою очередь, связана с модулями и дидактическими единицами, которые необходимо освоить обучающемуся. Модули и единицы покрываются областями дисциплин. Таким образом, в рамках режима работы *PBL* дисциплина не является основной единицей ОП, а служит формальным описанием к модулям, идентифицируя их принадлежность.

Также введем такое понятие, как «сжатый курс» – короткий интенсив, одной из целей которого является обобщение полученных образовательных результатов. Сжатый курс *связан* с дисциплиной через область знаний.

Все онтологии, несмотря на разницу в подходах, содержат пересекающиеся типы связей, которые было решено обобщить. Это позволило организовать хранение всех онтологий в единой базе данных. Перечень включает в себя связи:

- содержит (*rel*);
- формирует (*output*);
- пререквизит (*prereq*);
- иерархия (*hierarchy*);
- требует (*requires*);
- имеет / ссылается на (*has / refers*);
- проверяет (*check*);
- соответствует (*corresponds*);
- определяет (*defines*).

Для каждой онтологии были сформулированы свои «уровни детализации», в рамках которых онтология редуцировалась с помощью исключения сущностей, не требуемых на рассматриваемом уровне проектирования.

Например, рассмотрим режим работы *Backward Design*, когда необходимо составить только перечень компетенций с дисциплинами, которые их развивают. Тогда единственно необходимыми будут сущности «дисциплина», «компетенция» и связи «пререквизит», «формирует». Получившаяся онтология показана на рис. 4.

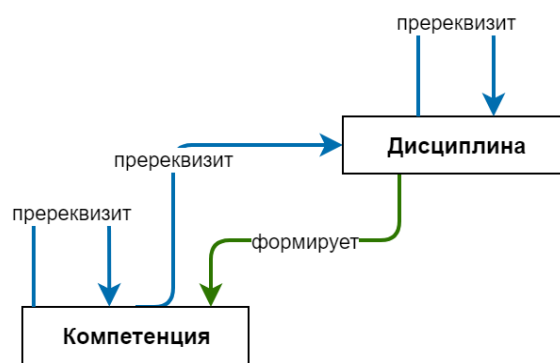


Рис. 4. Минимальная онтология для подхода *Backward Design*

АРХИТЕКТУРА СИСТЕМЫ, РЕАЛИЗУЮЩЕЙ РАССМАТРИВАЕМЫЕ ПОДХОДЫ.

Разработанные механизмы онтологий было необходимо реализовать в базе данных. Эта задача представляла сложность в виду необходимости хранить связи между объектами, а также между классами, разрешенными в рамках данной онтологии. Была реализован подход:

- типы отношений («пререквизит», «иерархия» и др.) хранятся в таблице *RelationsTypes*;
- информация о возможных связях между классами в рамках конкретных режимов работы вынесена в таблицу *RelationPermissions*; каждая ее строка содержит типы первого и второго компонентов, тип связи, разрешенной между ними, а также внешний ключ на режим работы, в рамках которого эта связь справедлива. Структура таблицы с примерами строк приведена на рис. 5;
- конкретная связь между двумя компонентами хранится в таблице *RelationObjects* и представляет собой кортеж, содержащий внешний ключ к объекту таблицы *RelationPermissions*, внешний ключ к объектам из таблиц первого и второго компонентов (определяются типами компонентов из *RelationPermissions*), а также внешний ключ на образовательную программу, для которой данная связь создается. Отметим, что в образовательной программе есть ссылка на режим проектирования, в рамках которого она разрабатывается. Структура таблицы с примерами строк приведена на рис. 6.

<i>id</i>	<i>first_component</i>	<i>second_component</i>	<i>rel_type</i>	<i>mode</i>
4	Дисциплина	Тема	содержит	ФГОС без ДЕ

Рис. 5. Структура таблицы *RelationPermissions*

<i>id</i>	<i>first_instance</i>	<i>second_instance</i>	<i>rel_perm</i>	<i>curriculum</i>
5	Программирование	Одномерные массивы	4	Программная инженерия

Рис. 6. Структура таблицы *RelationObjects*

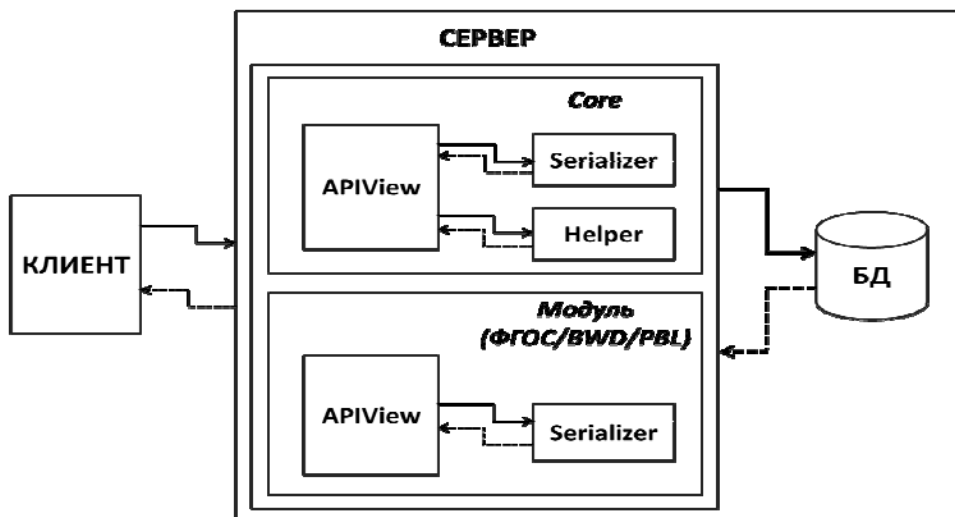


Рис. 7. Общая архитектура системы

Система спроектирована как веб-приложение, серверная часть которого состоит из главного (*core*) модуля, где хранится общая логика системы, и модулей, соответствующих трем подходам, описанным выше. Каждый модуль реализует *API* для доступа к своим данным и функциям, что позволяет гибко подходить к вопросу интерфейса системы. Общая архитектура системы показана на рис. 7.

ИСПОЛЬЗОВАНИЕ РАЗРАБОТАННЫХ ОБРАЗОВАТЕЛЬНЫХ ПРОГРАММ В КАЧЕСТВЕ БАЗЫ ЗНАНИЙ

Подход, предложенный в работе, позволяет пользователю при проектировании образовательного процесса опираться на опыт ранее разработанных программ. Они могут быть использованы как база знаний, которая способна автоматически отслеживать создаваемые новые программы и предлагать пользователю внести в них правки.

Был разработан перечень подсказок, описывающих возможность автоматического анализа проектируемой образовательной программы. Реализованные подсказки приведены в таблице.

Классификация подсказок системы при проектировании программ

Название	Определение
Подсказки компонентов и связей	При добавлении компонента система подсказывает пользователю компоненты и связи, соответствующие добавленному, в ранее построенных образовательных программах
Подсказки дублирования образовательных программ	При добавлении более N (число настраивается) компонентов в образовательную программу (даже без наличия связей между ними), пользователь получает информацию о ранее разработанных образовательных программах, в которых содержатся такие

Название	Определение
	же компоненты. С ростом N растет вероятность дублирования образовательной программы. Данная подсказка позволяет предотвратить подобные случаи
Подсказки уровней детализации	Информируют пользователя о недостающих компонентах и связях для данного уровня детализации образовательной программы. При этом демонстрируется шаблон образовательной программы согласно онтологии уровня детализации, описанного ранее
Подсказки третьего компонента	Подсказка уведомляет пользователя о существовании в другой ОП связи между двумя компонентами разрабатываемой образовательной программы через третий
Подсказки типов связей	Подсказки отображаются при попытке пользователя провести связи между компонентами, не предусмотренные в данном уровне детализации. Дают возможность переключить режим проектирования для расширения детализации

ЗАКЛЮЧЕНИЕ

Был разработан прототип системы образовательного проектирования, в котором реализована возможность гибкой настройки как подхода к созданию образовательных программ, так и уровня их детализации с помощью инструмента онтологий. Предложенные принципы хранения онтологии в базе данных позволяют гибко добавлять в систему формализацию новых подходов к образовательному проектированию.

СПИСОК ЛИТЕРАТУРЫ

1. Малкова И.Ю., Киселёва П.В. Образовательное проектирование в высшей школе: разработка проектов педагогической практики // Вестник Томского гос. ун-та. – 2011. – №346. – URL: <https://cyberleninka.ru/article/n/obrazovatelnoe-proektirovanie-v-vysshey-shkole-razrabotka-proektov-pedagogicheskoy-praktiki> (дата обращения: 02.09.2018).
2. ФГОС ВО (3++) по направлениям бакалавриата // Портал Федеральных государственных образовательных стандартов высшего образования. – 2017. – URL: <http://fgosvo.ru/fgosvo/151/150/24> (дата обращения: 02.09.2018).
3. Рекомендации по разработке и утверждению рабочих программ учебных курсов, предметов, дисциплин (модулей) / Городской методический центр. – URL: <http://mosmetod.ru/metodicheskoe-prostranstvo/srednyaya-i-starshaya-shkola/izo/metodicheskie-materialy/rekomendatsii-po-razrabotke-i-utverzheniyu-rabochikh-programm-uchebnykh-kursov-predmetov-distiplin-modulej.html> (дата обращения: 07.09.2018).
4. Генератор рабочих программ ТУСУР. – URL: <https://workprogram.tusur.ru/> (дата обращения: 07.09.2018).
5. LMS Moodle. – URL: <http://moodle.org> (дата обращения: 07.09.2018).
6. Porter G.W. Free choice of learning management systems: Do student habits override inherent system quality? // Interactive Technology and Smart Education. – 2013. – Т. 10, № 2. – С. 84-94.
7. Преимущества и недостатки системы дистанционного образования MOODLE. – URL: <http://eduros.ru/konf/infor/e9.html> (дата обращения: 07.09.2018).
8. Федеральные государственные образовательные стандарты. – URL: <http://минобрнауки.рф/документы/336> (дата обращения: 07.09.2018).
9. Лагерева А.В., Попков В.И., Горленко О.А. Компетентностный подход и ФГОС третьего поколения // Совершенствование инженерных образовательных программ. – 2012. – № 11. – С. 16–41.
10. Childre A., Sands J.R., Pope S.T. Backward design: Targeting depth of understanding for all learners // Teaching Exceptional Children. – 2009. – Т. 41, № 5. – С. 6-14.
11. Искренко Э.В., Полтон Т.А. Проблемно-ориентированное обучение: особенности методики преподавания в Великобритании (на примере St. George university of London, great Britain) // Научные ведомости Белгородского государственного университета. Серия: История. Политология. – 2008. – №10(50). – URL: <https://cyberleninka.ru/article/n/problemno-orientirovannoe-obuchenie-osobennosti-metodiki-prepodavaniya-v-velikobritanii-na-primere-st-george-university-of-london-great> (дата обращения: 07.09.2018).
12. The Science of Instruction and the Technology of Instructional Design. – URL: <http://mdavidmerrill.com/Papers/Reclaiming.PDF> (дата обращения: 07.09.2018).
13. Instructional Design. – URL: <http://educational-technology.net/instructional-design/> (дата обращения: 07.09.2018).
14. OWL 2 Web Ontology Language. RDF-Based Semantics (Second Edition): W3C Recommendation 11 December 2012. – 2012. – URL: <https://www.w3.org/TR/owl-rdf-based-semantics> (дата обращения: 15.10.2018).
15. Стюарт Р., Норвиг П. Искусственный интеллект: современный подход: 2-е изд. – М.: Вильямс, 2007. – 1408 с.

Материал поступил в редакцию 25.09.18.

Сведения об авторах

АБРАМСКИЙ Михаил Михайлович – старший преподаватель кафедры программной инженерии, Высшая школа информационных технологий и интеллектуальных систем (ИТИС), Казанский (Приволжский) федеральный университет (КФУ), Казань
e-mail: ma@it.kfu.ru

ЦИММЕРМАН Артур Михайлович – магистрант 2-го года, Высшая школа информационных технологий и интеллектуальных систем (ИТИС), Казанский (Приволжский) федеральный университет (КФУ), Казань
e-mail: arthurcots@gmail.com

АЛЬМУХАМЕТОВА Альбина Альбертовна – бакалавр программной инженерии, Высшая школа информационных технологий и интеллектуальных систем (ИТИС), Казанский (Приволжский) федеральный университет (КФУ), Казань
e-mail: aalmukhametova@gmail.com

АЛТЫНБАЕВА Диляра Тафкелевна – бакалавр программной инженерии, Высшая школа информационных технологий и интеллектуальных систем (ИТИС), Казанский (Приволжский) федеральный университет (КФУ), Казань
e-mail: altdilia@gmail.com