

Управление данными в современных цифровых образовательных средах

Статья рекомендована А.М. Елизаровым 15.01.2019.



АБРАМСКИЙ Михаил Михайлович
Старший преподаватель кафедры программной инженерии Высшей школы информационных технологий и интеллектуальных систем Казанского (Приволжского) федерального университета

Аннотация

Рассмотрены вопросы перехода на новые методы работы с данными для цифровых образовательных сред (ЦОС). Решается задача обеспечения гибкости проектирования компонентов ЦОС с целью их широкого и долгосрочного использования. Предложен подход, заключающийся в явном введении в ЦОС функционала управления форматами образовательных программ и цифровых портретов обучающихся, реализованный с помощью средств онтологического моделирования.

Ключевые слова:

цифровые образовательные среды, онтологии, цифровые портреты, модели данных, коллаборативная фильтрация, индивидуальные траектории.

Введение

Информационные образовательные системы (ИОС) уже в течение достаточно долгого времени являются необъемлемой частью современного образования. На сегодняшний день в сфере образования существует более 400 различных решений¹. Под влиянием требований цифровой экономики наблюдается переход к новому поколению ИОС — *цифровым образовательным средам*, представляющим собой комплексы интеллектуальных информационных решений, содействующих повышению эффективности, качества образования, индивидуализации, персонализации и адаптивности обучения.

Указ Президента РФ «О национальных целях и стратегических задачах развития Российской Федерации на период до 2024 года» ставит задачу «создания современной и безопасной цифровой образовательной среды, обеспечивающей высокое качество и доступность образования всех видов и уровней» [1]. В 2016 году Правительство РФ утвердило в рамках программы «Развитие образования» проект «Современная цифровая образовательная среда в РФ» [2]. Одновременно появляются и другие решения, претендующие на то, чтобы считаться ЦОС.

Цифровые образовательные среды наследуют главный недостаток ИОС — недостаточную гибкость работы с данными: жесткое задание используемых форматов, подходов, типов данных. Это приводит к снижению вариативности и нежеланию участников образовательного процесса, несмотря на высокую степень технологических навыков, пользоваться этими системами [3].

Недостаточная гибкость может быть объяснена использованием в большинстве решений *реляционной модели*, жестко задающей структуру базы данных (БД), изменять которую возможно только при программной доработке систем. Реляционные БД продолжают оставаться наиболее популярными, 9 самых используемых открытых систем управления обучением используют именно их².

В то же время отмечается [4], что ЦОС должны обеспечивать гибкость и возможность решать одни и те же задачи разными способами. Это требует иного подхода к представлению данных, включая возможность динамически моделировать

¹ 2018 Tech Landscape – <https://encoura.org/products-services/adventures-research-and-advisoryservices/higher-education-technology-landscape-2018>

² по данным сервиса подбора программного обеспечения capterra.com и портала elearningindustry.com, посвященного электронному обучению

типы и форматы, используя все преимущества аппарата баз знаний и нереляционных моделей данных.

В работе предложен подход к управлению данными наиболее значимых частей ЦОС — пользователей и образовательных программ. Обсуждены подходы к реализации в рамках предложенного подхода программной генерации индивидуальной образовательной траектории и механизмов рекомендаций, независимых от конкретных форматов. Освещены вопросы выбора архитектурного шаблона и модели данных для реализации предложенных подходов.

1. Иерархия School и Subschool

Большая часть работ по ЦОС посвящена построению в ней *индивидуальных образовательных траекторий* (ИОТ) обучающихся. Для этого в ЦОС должны присутствовать сведения об *образовательных программах* (ОП), *обучающихся* и *преподавателей*. Этот информационный массив содержит разные структуры, форматы, типы, классы и понятия. Подходы к содержанию одной и той же сущности могут различаться в рамках одной образовательной организации. Например, сведения о пользователе могут трактоваться как анкета или как граф компетенций. Необходима реализация возможности проектировать разные форматы работы с данными в рамках одной образовательной организации или ее подразделения. Для этого введены следующие понятия:

- **School** — структурная единица, реализующая образовательную деятельность, — факультет, институт, университет и др.
- **Subschool** — некий «регламент проектирования» форматов образовательных программ и сведений о пользователях. Примеры Subschool для института:
 - Образовательная программа магистратуры «Программная инженерия»;
 - Регламент дополнительного образования школьников;
 - Обучение компетенциям в области машинного обучения.

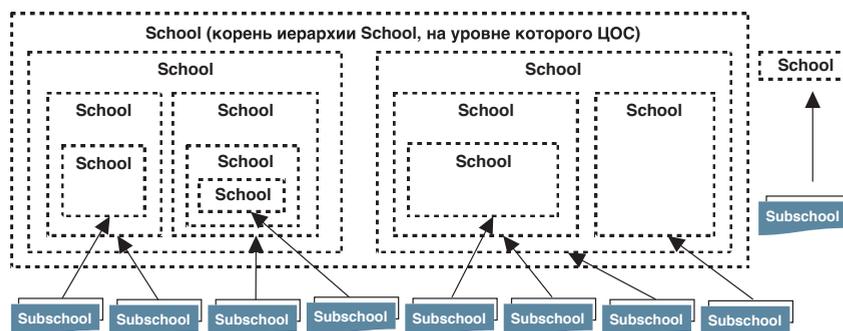


Рис. 1. Иерархия School и Subschool

Иерархия School и Subschool показана на рис. 1. В рамках School может быть создано несколько Subschool.

2. Моделирование обучающихся и преподавателей в ЦОС

Ключевыми ролями в ЦОС являются обучающиеся и преподаватели. Для их описания введено понятие цифрового портрета, представляющего собой данные о пользователе, хранимые в БД ЦОС, а также структуры представления этих данных. Цифровые портреты пользователей возникают в системах³, реализующих персонализацию своих ключевых функций. В рамках предлагаемого подхода цифровой портрет представлен набором элементов, каждый из них содержит следующие сведения:

- **Группа элементов цифрового портрета.** Введены 4 группы:
 - *Анкета:* персональные данные, аккаунты в социальных сетях и др.;
 - *Компетентностный портрет:* компетенции, знания, умения и навыки (ЗУН), пройденные предметы, курсы повышения квалификации, и переподготовки, онлайн-курсы, опыт работы и др.;
 - *Портфолио:* участия в олимпиадах, конкурсах, грантах, соревнованиях; грамоты, благодарственные письма, сертификаты;
 - *Личный портрет:* сведения о характере, типе личности, интересах; членство в организациях, объединениях.
- **Тип элемента.** Введены две категории типов:
 1. Типы, используемые в программировании: скалярные (числа, символы и булевый тип); дата и время; строковые типы и их частные случаи (e-mail, путь к файлу, URL и др.); файлы (медиа, документы и др.); списки значений перечисленных выше типов.
 2. Классы в смысле объектно-ориентированного программирования – наборы атрибутов. Каждый атрибут имеет название и значение типа первой или второй категории.
- **Перечень атрибутов** элемента, диктуемый его типом. Например, для элемента «Участие в конференции» атрибутами будут «Название», «Место проведения», «Дата проведения», «Результат».

Гибкость проектирования реализуется через возможность задавать перечень и поддерживаемое типов второй категории в разных Subschool. Предлагается проектировать

³ См., напр., «цифровой портрет пользователя-подростка в интернете» –<http://mgk.olimpiada.ru/work/9183/mrsl/>.

типы с помощью онтологий, зарекомендовавших себя как эффективный инструмент моделирования предметной области. В работе [5] онтологии использованы для моделирования навыков ИТ-специалистов, а в работах [6, 7] — для построения моделей компетенций.

Онтология описывает концептуальную схему предметной области, включающую классы, их экземпляры и связи между ними. Для задания онтологий был выбран подход *Resource Description Framework (RDF)* [8]. В нем онтологии задаются в виде списка триплетов вида **<Класс, Связь, Класс>** [9]. Триплеты могут трактоваться как ребра графа концептуальной схемы онтологии. Примеры триплетов:

: Олимпиада	: имеет	: Уровень
: Экзамен ЕГЭ	: имеет	: Оценка
: Компетенция	: имеет	: Степень освоения

Subschool может содержать до пяти онтологий: четыре — для проектирования типов каждой группы элементов и одну — для образовательных программ.

В рамках ЦОС тип «Компетенция» доступен во всех Subschool. Он содержит атрибуты: «*связанная компетенция ОП*» (для сопоставления цифрового портрета и ОП), «*формулировка*», «*тип*», «*степень освоения*» (семантика этого значения определяется экспертом School), «*входные требования*».

Для задания отношения «Класс–Атрибут» все онтологии цифровых портретов должны содержать тип связи «**Имеет**».

Цифровые портреты преподавателей строятся по аналогии. При построении ИОТ они учитываются при подборе учителей на предметы и под критерии соответствия обучающемуся. В онтологиях для этого добавлен тип «**Преподаваемый модуль**» с атрибутами «*связанный модуль ОП*», «*название*», «*тип модуля*».

3. Проектирование образовательных программ в ЦОС

Образовательная программа в рамках ЦОС представляет собой набор изучаемых модулей, формируемых компетенций, получаемых знаний, умений и навыков, используемых образовательных технологий и др.

Был проведен анализ типовых компонентов ОП. К ним отнесены *компетенции, модули, ЗУНы, проекты, темы, дидактические единицы, ресурсы, формы контроля* и т.п. Были также выделены типовые связи между компонентами ОП. Например, справедливы отношения «модуль формирует компетенцию», «компетенция относится к роли», «модуль содержит дидактическую единицу», «модуль А является пререквизитом модуля Б» и т.п.

Структура и содержание ОП в различных подходах и концепциях могут трактоваться по-разному. В высшем образовании в ОП включены рабочие программы дисциплин, календарный план, фонд оценочных средств [10]. Неакадемические, корпоративные ОП могут содержать лишь перечень модулей и компетенций. ОП в ЦОС могут быть созданы в разных концепциях с разной степенью детализации, что должно быть отражено при разработке соответствующих инструментов.

Образовательная программа может быть представлена как граф, где вершинами являются компоненты, а ребрами — связи между ними. В разных подходах

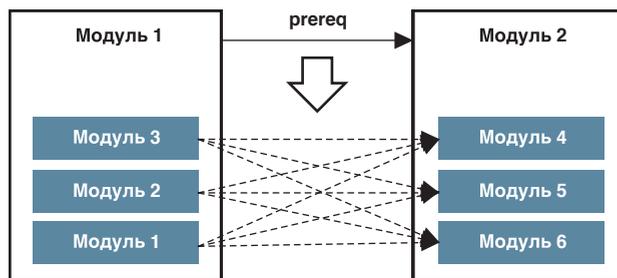


Рис. 2. Связь «Пререквизит» между модулями означает зависимость каждой темы модуля 1 от каждой темы модуля 2

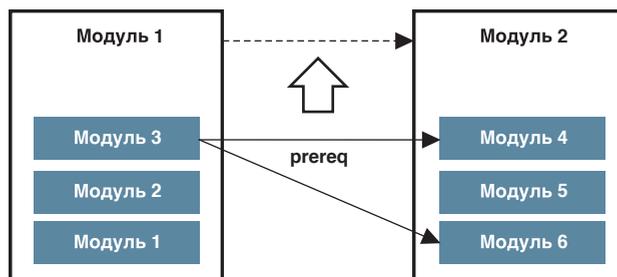


Рис. 3. Связи «Пререквизит» между некоторыми темами влекут наличие этой связи между модулями

внутри School возможны разные трактовки связей между одними и теми же компонентами. Предположим, что есть два **модуля**, при этом первый должен быть освоен обучающимся до прохождения второго. Каждый модуль **содержит темы**. Интерпретировать возможные связи «Пререквизит» между темами и модулями можно двумя способами:

1. Если модуль 1 – пререквизит модуля 2, то каждая тема модуля 2 должна зависеть от каждой темы модуля 1 (рис. 2);
2. Темы имеют собственную логику связей – на рис 3 темы T4 и T5 второго модуля зависят от темы T1 первого модуля, что означает наличие вычислимой связи «Пререквизит» между модулями 1 и 2.

Для моделирования разрешенных компонентов и связей по аналогии с цифровыми портретами были использованы онтологии. В каждой онтологии ОП обязательно наличие сущностей «Модуль» и «Компетенция» вместе с их входными требованиями, а также наличие триплетов:

: Модуль : пререквизит : Модуль
: Модуль : формирует : Компетенция

Эти сущности и связи являются инвариантными относительно любого подхода к организации образования [11].

Весь процесс проектирования ОП можно разбить на итерации создания одной связи (ребра графа). Порядок шагов следующий:

- создается объект **o1** класса **A**;
 - создается объект **o2** класса **B**;
- осуществляется попытка соединить **o1** и **o2** связью **C**; происходит запрос к онтологии ОП, проверяющий возможность такого соединения в ней; если это возможно, т.е. триплет $\langle A: C: B \rangle$ присутствует в онтологии, происходит сохранение ребра.

4. Коллаборативная фильтрация для компонентов ЦОС

В рамках предложенного подхода нужно описать характерные для ЦОС механизмы рекомендаций образовательных программ, модулей. Также реализация требуется для рекомендации потенциальных преподавателей и сокурсников, поскольку соответствующие исследования свидетельствуют о наличии влияния окружения (сокурсников и преподавателей) на образовательные результаты [12, 13].

Рекомендации в ЦОС были реализованы путем адаптации методов коллаборативной фильтрации [14], опирающихся на *бинарные меры сходства* [15]. В качестве наиболее подходящих выбраны меры Жаккара, Дайса и Отиаи (табл. 1). Компоненты ЦОС, для которых вычисляются данные меры, трактуются как множества — компетенций и характеристик (для цифровых портретов), компетенций и модулей (для образовательных программ).

Таб. 1. Выбранные для рассмотрения коэффициенты сходства

Коэффициент Жаккара	$\frac{ A \cap B }{ A + B - A \cap B }$
Коэффициент Дайса	$\frac{2 * A \cap B }{ A + B }$
Коэффициент Отиаи	$\frac{ A \cap B }{\sqrt{ A * B }}$

Для образовательных программ была выбрана мера Жаккара, для которой характерен рост значения в случае, когда большая часть одного множества входит в другое. Это схоже с задачами поиска похожих ОП — когда важны не симметричная похожесть одной программы на другую, а наличие в другой программе того же, что есть в первой.

При сравнении цифровых портретов не было выявлено принципиальной разницы между указанными мерами. Поэтому по умолчанию была использована мера Дайса, наиболее быстрая для вычисления. Любая мера соответствия определяется простой формулой, которую можно при необходимости реализовать в ЦОС.

С помощью выбранных мер были определены модели соответствия. Для образовательных программ таковыми являются «*схожесть по компетенциям и модулям*», «*схожесть по уровню входных требований*» и «*совпадение по входным требованиям*». Для цифровых портретов моделями соответствия являются «*схожесть*

по компетенциям», «по уровню компетенций и характеристикам», «различие по компетенциям и характеристикам», а также «соответствие обучению» — возможность преподавателю сформировать у обучающегося желаемую компетенцию.

5. Генерация индивидуальных образовательных траекторий

Индивидуальную образовательную траекторию можно трактовать как ОП, адаптированную для обучающегося на основе его цифрового портрета. Программу, выбранную для адаптации, будем называть *целевой*.

В научной статье группы авторов (Батырова Э. Ф., Марданова А. Р., Ахметзянова Т. А., Абрамский М. М.) [16] разработан алгоритм программной генерации индивидуального учебного плана, базирующийся на вычислении корреляции компетенций учебного плана, пожеланий абитуриента и требований профессиональных стандартов с получением на выходе списка рекомендованных дисциплин и компетенций. Предложенный алгоритм развивает идеи работы [16], реализуя требование автоматического учета требований к обучающемуся, его пожеланий, а также рекомендаций по компетенциям, модулям, возможным преподавателям и сокурсникам.

Индивидуальные образовательные траектории (ИОТ) можно представить следующим набором данных:

1. компетенции целевой ОП;
2. модули целевой ОП;
3. дополнительные компетенции, добавленные в траекторию;
4. дополнительные модули, добавленные в траекторию;
5. список рекомендованных преподавателей каждого модуля;
6. список рекомендованных сокурсников по данному модулю.

Рассмотрим алгоритм генерации ИОТ.

Шаг 1. Определение целевой ОП. Выбор целевой ОП, переход к шагу 2 (→ 2). Или выбор желаемых компетенций (осуществляется поиск наиболее подходящей ОП). Возможны варианты:

- **нашлись одна или несколько программ**, содержащих все компетенции; обучающийся делает выбор; → 2;
- **не нашлось** программы, целиком включающей желаемые компетенции; обучающемуся выводятся ОП, схожие со списком желаемых компетенций в терминах моделей соответствия. Возможен возврат к выбору компетенций или выбор одной из предложенных программ → 2;
На данном шаге возможны еще два независимых сценария:

- Развитие компетенций по рекомендациям — подбор модулей, развивающих компетенции, входящие в цифровой портрет;
- Рекомендация ОП на основе выбора, сделанного обучающимися со схожими цифровыми портретами.

Отметим, что желаемые компетенции, не вошедшие в программу, будут учтены на шаге 3.

Шаг 2. Проверка соответствия цифрового портрета требованиям ОП. Возможны следующие сценарии:

- а. Проверка вручную экспертом, который либо одобряет генерацию траектории на основе данной программы (→ 3) либо отклоняет ее (→ 1);
- б. Автоматическая проверка, сопоставляющая компетенции цифрового портрета и входные требования ОП. В портрете должны присутствовать элементы, соответствующие всем требованиям ОП, их степени освоения должны быть выше пороговых значений ОП. Типовой случай — унификация результатов экзаменов ЕГЭ и входных требований к образовательной программе; если требования соблюдены → 3, если нет → 1;

Шаг 3. Генерация модулей и компетенций ИОТ. В траекторию переносятся модули и компетенции целевой ОП. Если остались неучтенные компетенции, происходит поиск формирующих их модулей в других ОП иерархии School. Если портрет удовлетворяет входным требованиям найденных модулей, их можно добавить в ИОТ;

Если на шаге 1 была выбрана опция «Развитие компетенций по рекомендациям», осуществляется поиск модулей по иерархии School, требованиям которых удовлетворяет цифровой портрет. Также возможен поиск модулей, опирающийся на механизмы коллаборативной фильтрации по схожим портретам обучающихся. Найденные модули можно добавить в ИОТ.

Шаг 4. Рекомендация преподавателей и сокурсников. Для каждого модуля ИОТ происходит поиск преподавателей, в цифровом портрете которых отражен элемент, соответствующий возможности вести данный модуль. Также происходит поиск возможных преподавателей и однокурсников по выбранным моделям соответствия, настроенным заранее пользователем ЦОС с ролью «Эксперт».

6. Вопросы реализации предложенных подходов

Для реализации предложенных подходов в специальных программных инструментах в качестве базового архитектурного шаблона выбрана **микросервисная архитектура** [17]. В ней приложение представляет собой совокупность *слабовязанных* сервисов, взаимодействующих по протоколу HTTP и имеющих возможность независимого развертывания. Микросервисная архитектура позволяет реализовывать

сервисы с помощью разных языков программирования и технологий, тогда в разработке может участвовать больше квалифицированных специалистов [18].

Для снижения риска сильной связности сервисов из-за работы с общими данными БД была реализована как отдельный сервис. Для работы других сервисов с сервисом БД используется программный интерфейс (API), основанный на REST-архитектуре. При разработке новых сервисов необходимо ориентироваться на существующий API, по необходимости уточняя его. Для описания API использована спецификация OpenAPI⁴, для генерации API — инструмент Swagger⁵, позволяющий описывать API на языке разметки YAML⁶.

Еще одним важным вопросом является выбор модели данных. Жесткость реляционной модели данных в силу онтологического моделирования типов компонентов ЦОС больше не является ограничением, ее применимость для реализации предложенных подходов исследовалась наравне с **графовыми БД**, относящимися к классу нереляционных [19]. Самым популярным инструментом создания графовых БД является СУБД *Neo4j*⁷, где для работы с графами используется язык *Cypher*⁸. В графовой БД вершины сгруппированы по видам с помощью специальных меток, которые означают принадлежность вершины какому-то классу. В метках ребер также можно хранить данные. Графовые БД не имеют понятия «схемы», т.е. заранее определенных типов и их ограничений, классы-метки и типы ребер появляются по мере добавления данных.

В типовых запросах к ЦОС реляционные и графовые модели работают с одинаковой вычислительной сложностью. Однако для некоторых задач запросы языка *Cypher* для *Neo4j* выигрывают по сравнению с реляционной моделью.

Пусть для модуля **М** нужно получить все предшествующие модули, необходимые для его изучения. Это означает, что необходимо сначала найти модули, связанные с **М** связью "пререквизит", затем рекурсивно для каждого найденного модуля найти предшествующие ему, вплоть до начальных модулей ОП.

Запрос в реляционных БД требует многократного обращения к таблице связи «пререквизит», поскольку для очередного модуля ведется поиск его предшественников. Запрос для графовых БД просто обходит граф связи «пререквизит», что не требует для каждого шага нового обращения к списку модулей, поскольку в графовых БД в явном виде реализованы ссылки между вершинами.

Для оценки трудоемкости запросов в обеих моделях были использованы открытые статистические данные ряда ведущих вузов России. Установлено, что запрос для этой задачи в графовых БД на несколько порядков быстрее реляционного, и разница увеличивается с ростом количества ОП и связей «пререквизит».

Таким образом, для реализации работы с образовательными программами, предпочтительно использовать графовые БД. В то же время для других данных, запросы к которым не требуют реализации рекурсивных операций, может быть применена реляционная модель.

⁴ The OpenAPI Specification Repository - <https://openapis.org/>

⁶ The Official YAML Web Site - <http://yaml.org/>

⁸ Cypher Query Language - <https://neo4j.com/developer/cypher-query-language/>

⁵ Swagger - The Best APIs are Built with SwaggerTools - <https://swagger.io/>

⁷ Neo4j Basics - <https://neo4j.com/product/>

Заключение

Описан подход, позволяющий в рамках ЦОС одной образовательной организации реализовать разные форматы и подходы на уровне хранения данных для своих подразделений. Предложены способы моделирования цифровых портретов и образовательных программ с помощью средств онтологий.

Показана возможность единого подхода к генерации индивидуальных образовательных траекторий в рамках одной ЦОС. Механизм рекомендаций реализован с помощью методов коллаборативной фильтрации.

В дальнейших исследованиях предполагается, с одной стороны, совершенствование форм представления компонентов ЦОС и программных инструментов работы с ними, с другой стороны, планируется развивать исследования по трансформации цифровых решений в образовании в сторону микросервисной архитектуры.

Также планируются дальнейшие исследования возможностей нереляционных БД в образовании, а также разработка инструментов, позволяющих создавать приложения с одновременным использованием нескольких моделей данных.

ЛИТЕРАТУРА

- О национальных целях и стратегических задачах развития Российской Федерации на период до 2024 года:** Указ Президента Российской Федерации № 204 от 07.05.2018. 2018. URL: <http://kremlin.ru/events/president/news/57425> (дата обращения: 15.01.2019).
- Современная цифровая образовательная среда в РФ.** — URL: <http://neorusedu.ru/> (дата обращения: 15.01.2019).
- ECAR Study of Undergraduate Students and Information Technology,** 2017. — 2017. — URL: <https://library.educause.edu/resources/2017/10/ecar-study-of-undergraduate-students-and-information-technology-2017> (дата обращения: 15.01.2019).
- КУШНИР М. **Цифровая образовательная среда. 2017.** URL: <https://medium.com/direktoria-online/the-digital-learning-environment-f12550b942a> (дата обращения: 15.01.2019).
- КУРЗАЕВА Л.В., ЧУСАВИТИНА Г. Н., МУСИЙЧУК М. В. **Разработка базы знаний интеллектуальной системы поддержки обучения ИТ-специалистов с использованием онтологического моделирования** // Интернет-журнал «Мир науки». 2017. Т 5. № 6. URL: <https://mir-nauki.com/PDF/99PDMN617.pdf> (дата обращения: 15.01.2019).
- PAQUETTE G. **An ontology and a software framework for competency modeling and management** // Educational Technology & Society. 2007. V. 10. No 3. P. 1-21.
- ЗИНДЕР Е.З., ТЕЛЬНОВ Ю. Ф., ЮНАТОВА И. Г. **Методика построения модели компетенций на основе профессиональных стандартов в области ИКТ для создания программ дополнительного профессионального образования** // Статистика и экономика. 2011. № 6(2). С. 112-118.
- RDFA Core 1.1 – Third Edition: W3C Recommendation 17 March 2015.** URL: <https://www.w3.org/TR/rdfa-core> (дата обращения: 15.01.2019).
- RDF 1.1 N-Triples. A line-based syntax for an RDF graph:** W3C Recommendation 25 February 2014. URL: <https://www.w3.org/TR/n-triples> (дата обращения: 15.01.2019).
- ФГОС ВО (3++) по направлениям бакалавриата: Портал Федеральных государственных образовательных стандартов высшего образования. 2017.** URL: <http://fgosvo.ru/fgosvo/151/150/24> (дата обращения: 15.01.2019).
- АБРАМСКИЙ М.М., ТИМЕРХАНОВ Т. И. **Сравнительный анализ использования реляционных и графовых баз данных в разработке цифровых образовательных систем** // Вестник Новосибирского государственного университета. 2018. Т. 16. № 4. С. 5-12.
- REEVES P. M., PUN W. H., CHUNG K. S. **Influence of teacher collaboration on job satisfaction and student achievement** // Teaching and Teacher Education. 2017. Т. 67. P. 227-236.
- WENTZEL K. R., RAMANI G. B. (ED.). **Handbook of social influences in school contexts: Social-emotional, motivation, and cognitive outcomes.** Routledge. 2016. 452 p.
- SU X., KHOSHGOFTAAR T. M. **A survey of collaborative filtering techniques** // Advances in artificial intelligence. 2009. Т. 2009. 19 p.
- CHOI S.S., CHA S. H., TAPPERT C. C. **A survey of binary similarity and distance measures** // Journal of Systemics, Cybernetics and Informatics. 2010. Т. 8. № 1. P. 43-48.
- АБРАМСКИЙ М.М., БАТЫРОВА Э. Ф., МАРДАНОВА А. Р., АХМЕТЗЯНОВА Т. А. **Генерация индивидуальных образовательных траекторий и расписания обучения в парадигме индивидуализации образования** // Электронные библиотеки. 2018. Т. 21. № 3-4. С. 129-145.
- Microservices** // James Lewis, Martin Fowler. 2014. URL: <https://martinfowler.com/articles/microservices.html> (дата обращения: 15.01.2019).
- RICHARDSON, C. **Pattern: Microservice Architecture** / C. Richardson. 2014. URL: <http://microservices.io/patterns/microservices.html> (дата обращения: 15.01.2019).
- MAKRIS A. **A classification of NoSQL data stores based on key design characteristics** // Procedia Computer Science. 2016. No 97. P. 94-103.