

Казахский национальный исследовательский технический университет  
имени К. И. Сатбаева

УДК 004.82

На правах рукописи

**УТЕГЕНОВА АНАР УРАНТАЕВНА**

**Модели и методы представления и организации образовательных ресурсов  
многоуровневой системы подготовки специалистов на основе онтологии**

6D070400 -Вычислительная техника и программное обеспечение

Диссертация на соискание ученой степени  
доктора философии (PhD)

Научный консультант:  
д.т.н., проф. Ахметов Б.С. ,  
к.т.н. доц. Кубеков Б.С.

Зарубежный консультант:  
профессор Дитмар Байер,  
Университет прикладных наук,  
Германия

Республика Казакстан

Алматы, 2018

## СОДЕРЖАНИЕ

|          |  |     |
|----------|--|-----|
|          | <b>ВВЕДЕНИЕ</b>  | 10  |
| <b>1</b> | <b>КОНЦЕПЦИЯ ПАРАДИГМЫ И ФОРМАЛИЗМЫ МОДЕЛИРОВАНИЯ ЗНАНИЙ</b>                                     | 14  |
| 1.1      | Обзор информационных технологий в образовательной практике                                       | 14  |
| 1.2      | Анализ существующих формализмов моделирования понятий и адаптация к требованиям                  | 19  |
| 1.3      | Инженерия образовательной предметной области   | 25  |
| 1.4      | Формализмы моделирования характеристик   | 25  |
| 1.5      | Визуализация онтологии и характеристической модели   | 36  |
| <b>2</b> | <b>КОНЦЕПЦИЯ СДИО И ПРОЕКТНО - ОРИЕНТИРОВАННАЯ ОБРАЗОВАТЕЛЬНАЯ СРЕДА</b>                         | 40  |
| 2.1      | Компетентностный подход  | 40  |
| 2.2      | Проектно-ориентированное инженерное обучение по направлению "Инженерия программного обеспечения" | 46  |
| 2.3      | Компетентностные модели и этапы СДИО   | 48  |
| 2.4      | Моделирование образовательных траекторий обучения  | 52  |
| <b>3</b> | <b>АДАПТИВНАЯ ОБРАЗОВАТЕЛЬНАЯ СРЕДА</b>  | 59  |
| 3.1      | Задачи адаптивной образовательной среды  | 59  |
| 3.2      | Проектные решения задач образовательной среды  | 65  |
| 3.2.1    | Формирование образовательных (знаниевых) компонент планируемого обучения                         | 65  |
| 3.2.2    | Проектирование дисциплин специальности   | 69  |
| 3.2.3    | Структурирование учебного материала дисциплины   | 75  |
| <b>4</b> | <b>ПОИСКОВАЯ СИСТЕМА ПРОЕКТНО – ОРИЕНТИРОВАННОЙ СРЕДЫ</b>  | 82  |
| 4.1      | Проектные решения, создание микросервисов  | 82  |
| 4.2      | Моделирование поисковой системы  | 85  |
| 4.2.1    | Варианты использования поисковой системы   | 85  |
| 4.2.2    | Диаграммы вариантов использования  | 86  |
| 4.2.3    | Моделирование динамических аспектов поисковой системы  | 88  |
| <b>5</b> | <b>СЦЕНАРИИ РАБОТЫ ОБРАЗОВАТЕЛЬНОЙ СРЕДЫ</b>   | 92  |
| 5.1      | Сценарий работы эксперта   | 92  |
| 5.2      | Сценарий работы преподавателя  | 103 |
| 5.3      | Экспериментальные результаты   | 106 |
| 5.4      | Метод исследования   | 106 |
| 5.5      | Результаты сравнительного анализа облачных сервисов по выявлению концептов                       | 108 |
|          | <b>ЗАКЛЮЧЕНИЕ</b>  | 111 |
|          | <b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ</b>  | 115 |
|          | <b>ПРИЛОЖЕНИЕ А</b> Листинг программного интерпретатора  | 121 |
|          | <b>ПРИЛОЖЕНИЕ Б</b> Сценарий работы эксперта со средой   | 133 |

## НОРМАТИВНЫЕ ССЫЛКИ

В настоящей диссертации использованы ссылки на следующие нормативные документы:

ISO/IEC 15288:2008. Systems and software engineering - System life cycle processes ([www.iso.org](http://www.iso.org));

ISO 19150-2, Geographic information - Ontology -Part 2: Rules for developing ontologies in the Web Ontology Language (OWL);

ISO 19103:-1), Geographic information - Conceptual schema language;

ISO 19107:2003, Geographic information - Spatial schema;

ISO 19108:2002, Geographic information - Temporal schema;

ISO 19109:-2), Geographic information - Rules for application schema;

ISO 19112:2003, Geographic information - Spatial referencing by geographic identifiers;

W3C OWL 2, OWL 2 Web Ontology Language: Structural Specification and Functional-Style Syntax (W3C Recommendation 27 October 2009);

W3C OWL 2 RDF, OWL 2 Web Ontology Language RDF-Based Semantics (W3C Recommendation 27 October 2009).

## ОПРЕДЕЛЕНИЯ

В настоящей диссертации применяют следующие термины с соответствующими определениями:

**Выражение знания** - спецификация онтологии опорного понятия домена обучения, операндами которого являются идентифицирующие и конкретизирующие понятия, между которыми определены отношения «композиция», «агрегация» и «альтернативный выбор». спецификация сетевой модели отображения знания, в виде выражения знания, составляющих базу знаний CDIO Syllabus.

**Выражение спецификации** - спецификация опорного понятия семантического контекста домена обучения в виде характеристической модели, представляющей собой иерархию идентифицирующих и конкретизирующих характеристик, между которыми определены отношения «композиция», «агрегация» и «альтернативный выбор».

**Декларативные знания** - это знания о мире задачи, то есть знания, описывающие свойства предметной области, в которой решается поставленная задача.

**Инженерия образовательной области** - совокупность методов анализа образовательной области, с целью определения семейств образовательных ресурсов и их систематизации, в форме рабочих средств многократного применения.

**Домен обучения** (фр. *domaine* - область, единица структуры) - когнитивная область знаний, в пределах которой наиболее точно и конкретно выявляются семантика и значение входящих в эту область понятий, фраз или совокупности фраз.

**Знание** - зафиксированная и проверенная практикой информация, которая может многократно использоваться людьми для решения тех или иных задач.

**Знаниевый компонент** – структурная единица базы знаний CDIO Syllabus, представляющая собой спецификацию онтологии опорного понятия в виде выражения знания.

**Знаниевая модель** - необходимый и достаточный набор взаимосвязанных онтологий опорных понятий домена обучения, вместе с их свойствами, характеристиками и причинно-следственными связями, а также классификацией опорных понятий по типам, ситуациям и признакам.

**Идентифицирующие понятия** – понятия первого уровня онтологии опорного понятия, с помощью которых определяются семантические и отличительные свойства опорного понятия, относительно других опорных понятий домена обучения.

**Инженерия домена обучения** - методика анализа и формализации знаний домена обучения, с целью разработки на их основе знаниевых компонент и их повторного использования.

**Информация** - данные, сопровождающиеся смысловой нагрузкой, помещенные в некоторый контекст и каким-либо образом оцениваемые получателем информации.

**Компетентность** - ядро профессионализма, на котором вырастает мастерство профессионала.

**Компетенция** - динамичная совокупность знаний, умений, навыков и способностей выпускника, которая необходима для его будущей профессиональной деятельности и личностного развития, и которую выпускник должен освоить и продемонстрировать после завершения части или всей образовательной программы.

**Компетентностная модель** - необходимый и достаточный набор онтологий опорных понятий предметной области.

**Компетентностная модель этапа CDIO** - композиция профессиональных, базовых и дополнительных компетенций, которыми должны владеть обучающиеся, для успешного выполнения заданий и практических работ этапа CDIO.

**Компетентностный подход** - инструмент усиления социального диалога высшей школы с миром труда, и средство углубления их сотрудничества в условиях взаимного доверия.

**Конкретизирующее понятие** – понятие второго уровня онтологии опорного понятия, которое, в контексте своего идентифицирующего понятия, конфигурирует семантические и отличительные свойства опорного понятия, с помощью типичных, либо всевозможных сочетаний дочерних понятий, обладающих ясным и недвусмысленным описанием опорного понятия.

**Онтология** - подробная спецификация понятийной (терминологической) структуры семантического контекста домена обучения образовательной области.

**Онтология опорного понятия** - иерархия из двух уровней понятий, корневой вершиной которой является опорное понятие. Первый уровень составляют идентифицирующие понятия, контекстом которых является опорное понятие, второй уровень - конкретизирующие понятия, контекстом которых являются идентифицирующие понятия. На понятиях онтологии определены свойства общности и изменчивости, а также отношения «композиция», «агрегация» и «альтернативный выбор».

**Образовательные технологии** - неотъемлемая часть содержания обучения, связанная с формами представления, организации и генерирования знаний, на основе информационных технологий.

**Образовательная траектория обучения** - некоторая структура из знаниевых компонент обучения, освоение которых должно обеспечивать технологические процессы разработки и производства инженерных объектов и систем.

**Опорное понятие** - это базовая абстракция семантического контекста домена обучения, на базе которого формируется онтология.

**Понятие** (concept) - это любая мысль, в которой отражаются общие существенные свойства и связи предметов и явлений семантического контекста домена обучения.

**Парадигма** - способ построения системных абстракций, на основе свойств общности и изменчивости.

**Парадигма образования** (от греч. paradeigma - пример, образец) - базовая модель конкретного способа организации учебной информации, на основе свойств общности и изменчивости и, в целом, как ведущий подход к выбору содержания и форм организации образовательных ресурсов.

**Предикат** – логическая N-арная пропозициональная функция, определенная для предметной области и принимающая значения либо истинности, либо ложности.

**Предметная область** – это совокупность знаний, значимая для пользователей и, соответственно, область специализации или интереса, внутри которых находятся семейства целых систем.

**Проектно-ориентированная образовательная среда** - компонентная система, в которой осуществляется разработка и использование повторно используемых знаниевых компонент, для проектирования образовательных программ и организации учебного процесса.

**Пропозициональное представление** – внутреннее представление мысли, отличное от представления на естественном языке.

**Репозиторий** учебно-методических материалов проектно-ориентированной образовательной среды – хранилище знаниевых компонент и другой учебно-методической информации по дисциплинам специальности.

**Родительское понятие** – абстрактное понятие, выражающее семантическую общность для своих дочерних понятий.

**Семантический анализ домена обучения** – процессы выделения семантических контекстов области знания и структурирования их системами опорных и конкретизирующих понятий.

**Семантический контекст домена обучения** - некоторое информационное (образовательное) пространство, в пределах которого любое опорное понятие отображает семантическую общность и имеет индивидуальные свойства, присущие данному пространству.

**Smart- обучение** - гибкое обучение, предполагающее наличие большого количества источников знаний, максимального разнообразия мультимедиа (аудио, видео, графика), способности быстро и просто настраиваться под уровень и потребности слушателя с помощью Smart – технологий и Smart – устройств.

**CDIO** - инициатива MIT по инженерному образованию, основанная на методе проектного обучения, в котором этапами разработки проекта являются: Conceiving, Designing, Implementing и Operating.

**CDIO Standards** – стандарты инициативы MIT, в которых конкретные результаты обучения представлены в виде подробного перечня компетенций или квалификационных требований инженера.

**CDIO Syllabus** – учебно-методическое ядро проектно-ориентированной образовательной среды, формируемое из знаниевых компонент, назначением которого является обеспечение обучающихся суммой профессиональных знаний по разработке проекта на всех стадиях инициативы CDIO, а также предоставление необходимого и достаточного контента знаний, для проектирования дисциплин учебного плана специальности.

**Формализация** - метод исследования, в основе которого лежит отображение содержательного знания в знаково - символическом виде (формализованном языке).

**Характеристика** - различимое существенное свойство, а также значимый и доступный аспект понятия.

**Характеристическая модель** - модель с обобщением характеристик и изменяемых параметров понятия, а также логическим обоснованием композиции и агрегации характеристик, для количественной спецификации свойств понятия.

**Экземпляр опорного понятия онтологии** – упорядоченная последовательность дочерних понятий опорного понятия онтологии, связанных друг с другом отношениями "композиция", "агрегация" и "альтернативный выбор".

**Эпистемологическая функция компетентностного подхода** включает в себя: усиление ориентированности образования на трудоустройство, то есть на повышение конкурентоспособности выпускников вуза на рынках труда; на увеличение гибкости высшего образования, на основе обеспечения изоморфизмов знаниевого (предметно-дисциплинарного) и компетентностного подходов на учебно-модульной основе; повышение измеримости качества выпускников вуза и обеспечение совместимости результатов подготовки выпускников по одному профилю профессиональной деятельности в разных вузах и разных странах.

## ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

В настоящей диссертации применяются следующие обозначения и сокращения:

CDIO- всемирная инициатива по инженерному образованию, основанная на методе проектного обучения и состоящая из четырех этапов: *Conceiving* - этап анализа предметной области и моделирования артефактов приложения; *Designing* - этап проектирования артефактов приложения; *Implementing* - этап реализации, кодирования и оценки качества артефактов приложения и, наконец, *Operating* - этап управления и сопровождения приложения.

КМэ - компетентностная модель этапа CDIO, которая представляет собой композицию необходимых и достаточных наборов опорных понятий соответственно профессиональных, базовых и дополнительных компетентностей этапа. Другими словами, компетентностная модель этапа CDIO, с одной стороны, определяет семантические знания этапа, а с другой - модель управления учебно-познавательной деятельностью обучающихся.

ДКМэ - дескриптор компетентностной модели этапа CDIO, который содержит описание компетентностей обучающихся, при условии успешного освоения семантических знаний, необходимых для разработки проекта.

ПК.КМэ - профессиональные компетенции компетентностной модели этапа КМэ.

БК.КМэ - базовые компетенции компетентностной модели этапа КМэ.

ДК.КМэ - дополнительные компетенции компетентностной модели этапа КМэ.

ЕК&ontology - выражения знаний и онтологии опорных понятий профессиональных, базовых и дополнительных компетенций компетентностной модели этапа.

CD - дескриптор дисциплины, определяющий всю необходимую информацию по дисциплине.

КС (Knowledge Components) - образовательный (знаниевый) компонент, представляемый в виде выражения знания и его онтологии. Любое понятие в выражении знания играет роль ссылки на соответствующую информацию об этом понятии в репозитории среды.

КС.Syllabus.Base - множество образовательных (знаниевых) компонент планируемого обучения CDIO Syllabus, используемое для проектирования базовых дисциплин учебного плана.

КС.Syllabus.Prof - множество образовательных (знаниевых) компонент планируемого обучения CDIO Syllabus, используемое для проектирования профилирующих дисциплин учебного плана.

API – это единая точка входа среды, позволяющая просматривать статьи и осуществлять поиск.

SEO – это поисковая оптимизация (search engine optimization, SEO) - комплекс мер для поднятия позиций сайта, в результатах выдачи поисковых систем, по определенным запросам пользователей.



MVC – это схема разделения данных приложения, пользовательского интерфейса и управляющей логики на три отдельных компонента: модель, представление и контроллер таким образом, что модификация каждого компонента может осуществляться независимо.

WEB – это распределенная система, предоставляющая доступ к связанным между собой документам, расположенных на различных компьютерах, подключенных к Интернету.

## ВВЕДЕНИЕ

Тема диссертации «Модели и методы представления и организации образовательных ресурсов многоуровневой системы подготовки специалистов на основе онтологии». В диссертации исследовались вопросы разработки моделей представления и организации знаний с использованием онтологического подхода. В работе освещены основные этапы разработки методики проектирования учебных планов на основе метода проектного обучения и концепций всемирной инициативы CDIO.

Основная концепция диссертационного исследования предполагает комплексную модернизацию всех образовательных процессов, а также методов и технологий, используемых в этих процессах. Фактически, в области образования, модели и положения диссертации предусматривают переход к новой образовательной парадигме представления и организации знаний, а также в совершенствовании управленческого механизма, путем активного планирования и реализации образовательных программ и процессов, доступных широкому кругу учебных заведений Республики Казахстан.

В области образования основные положения диссертации предусматривают переход к новой образовательной парадигме представления и организации знаний, в совершенствовании управленческого механизма путем форсайтного планирования и реализации образовательных программ и процессов, доступных широкому кругу высших учебных заведений Республики Казахстан.

**Актуальность исследования** заключается в:

- поддержке процедур правительства Республики Казахстан по реализации Государственной программы по развитию науки и образования на 2016-2019гг., и программы по цифровизации отраслей экономики;
- предоставлении новой парадигмы информационной базы обучения, отвечающей потребностям цифровой экономики, с акцентом на развитие навыков в анализе информации и креативности мышления;
- диверсификации методик электронной педагогики и ассортимента образовательных услуг за счет освоения новых образовательных, в том числе Smart-технологий, с целью повышения эффективности и качества обучения;
- в модернизации учебных программ и планов подготовки инженерных кадров, с целью развития эпистемологической функции компетентностного подхода;
- в необходимости применения методов инженерии знаний и онтологического инжиниринга (рисунок Б.2), а также внедрения информационных технологий и педагогических средств интенсификации учебно-познавательной деятельности обучающихся.

Выявлены новые области научных проблем, которые должны применяться конкретно для Казахстана, модернизация содержания высшего и послевузовского образования в контексте мировых тенденций. В настоящее время большую роль оказывает уровень образования и объем знаний,

накопленных обществом. Особую роль в этом играют информационные технологии, а в них методы и средства работы со знаниями. Настоящие проблемы и пути их реализации представлены в процедурах правительства Республики Казахстан, Государственной программы по развитию науки и образования на 2016-2019гг., и Государственной программой «Цифровой Казахстан».

Программа включает проблемы сохранения и повторного использования знаний, традиционные для направления инженерия знаний. Эффективные рекомендации по реализации управленческих задач с использованием знаний исследовались С. Н. Дегтяревым [1]. В трудах О. М. Топоркова, сформулированы системы онтологий и описаны методы работы со знаниями [2]. Труды И. Ю. Шполянской, Л. Р. Черняховской посвящены вопросы качества электронного обучения [3, 4]. Исследования С. В. Титенко отражают методы анализа естественно-языковых текстов [5]. В трудах А.Е. Ермакова, рассматривается автоматизация онтологического инжиниринга, приводятся методы проектирования знание-ориентированных информационных систем а также описывается представление и прикладная обработка технологии извлечения предметно-ориентированных знаний [6]. Представление онтологической модели представлено в трудах Н.И. Цукановой [7], Т.А. Гавриловой [8] и других. Теория Д.А. Поспелова о семиотическом моделировании, представляет комплекс понятий через их связь между собой. Особое внимание заслуживает исследования Е.М. Бениаминова по проблемам, внедрения и использования библиотек онтологий [9]. Интерес к изучению проблем образования нашел свое отражение в многочисленных исследованиях казахстанских ученых: Мутанова Г.М.[10], Ахметова Б.С.[11], Саданова Б.М.[12], Шарипбаева А.А.[13], Кубекова Б.С.[14-25].

**Научная новизна** заключается в исследовании структуры формального представления данных и разработке синтаксиса языка спецификации знаний для глубокой модернизации образовательных программ в области техники и технологий, а так же моделирование и проектирование методик компетентностного подхода в практической плоскости проектного обучения Всемирной инициативы CDIO, метода форсайтных исследований и онтологического инжиниринга.

**Цель диссертации** заключается в разработке интеллектуального сервиса и модели представления и организации знаний с использованием компетентностного и онтологического подходов; разработке методики проектирования дисциплин и учебных планов многоуровневой системы, основанных на проектном методе обучения и концепциях Всемирной инициативы CDIO.

**Для реализации целей диссертации поставлены следующие задачи:**

1. Исследование методик повышения эффективности педагогических и информационных технологий, по развитию эпистемологической функции компетентностного подхода.

2. Исследование эмпирических моделей формализации и отображения знаний образовательной области, моделей спецификации знаний на основе онтологического подхода, технологий и инструментальных средств Smart-образования.

3. Разработка методики применения онтологического подхода для проектирования онтологий учебного контента дисциплин и знаниевых компонент планируемого обучения образовательной среды.

4. Организация и представление моделей знаний и формирование знаниевых компонент планируемого обучения CDIO Syllabus,.

5. Разработка и обоснование проектных и инфраструктурных решений образовательной среды.

6. Экспериментальная проверка эффективности функционирования образовательной среды.

**Объектом исследования** Модели, методы и программные средства образовательных ресурсов.

**Предмет исследования** методика формирования образовательного контента на основе онтологии и управление образовательными программами и процессами.

**Ожидаемые результаты:** В области управления образовательными программами и процессами предполагается внедрить новые методики онтологического инжиниринга учебного контента для проектирования дисциплин и учебных планов специальности. Будут разработаны новые инструментальные средства и методики Smart-обучения, с учетом компетентностных моделей этапов CDIO и проектного метода обучения для ВУЗов Республики Казахстан. Предложенные новые решения аргументированы экспериментальной проверкой когнитивной способности обучающихся и апробированы в реальных условиях [14-25].

**Применимость полученных научных результатов:** Предполагается распространение результатов и рекомендаций проекта в плане изменения структуры образовательных программ и процессов, в рамках проекта по теме «Исследование и разработка моделей и методики представления и организации знаний с применением онтологического подхода и инструментальных средств Smart-технологии, при реализации образовательных программ и процессов», руководитель к.т.н., доцент РГП на ПХВ "Институт информационных и вычислительных технологий" КН МОН РК, Кубеков Б.С..

**Целевые потребители полученных результатов:** Результаты диссертационной работы ориентированы на использование в высших учебных заведениях для оптимального формирования учебных планов и программ.

В качестве программного продукта выступает Web- приложение на платформе ASP.NET, которое ориентировано на индивидуализацию траекторий обучения с точки зрения предпочтений и индивидуальных возможностей обучающихся, подготовку обучающихся к успешной профессиональной деятельности в условиях цифрового общества и умной экономики. Диссертация, имеет прикладное значение, приводятся сведения о практическом

использовании научных результатов, подтвержденные актами внедрения, имеющие теоретические рекомендации по использованию научных выводов.

**Методы исследования.** В диссертационной работе применяются методы онтологического анализа образовательной области; формальные системы и процедуры вывода знаний, сетевая и алгебраическая модели отображения знаний; концепции и механизмы порождающего и объектно-ориентированного программирования.

Разработанные модели и алгоритмы реализованы в виде программной системы и внедрены в деятельность РГП на ПХВ "Институт информационных и вычислительных технологий" КН МОН РК. Практическое использование результатов диссертационной работы подтверждено соответствующими документами.

**Основные положения и результаты исследований** обсуждались на учебно-методических и научных семинарах кафедры "Компьютерная и программная инженерия" университета "Туран", а так же на одноименной кафедре Казахского национального исследовательского технического университета им. Сатпаева К.И., на семинарах в РГП на ПХВ "Институт информационных и вычислительных технологий" КН МОН РК, и выступлениях на международных научно-практических конференциях. Основные научные результаты диссертации изложены в: 21 публикации, в том числе 5 в научных изданиях, рекомендуемых КН МОН РК, 5 в международных научных изданиях, входящих в базу данных Scopus, 11 в материалах международных научно-практических конференций.

**Основания для выполнения работы.** Диссертационная работа выполнялась в рамках проекта «Исследование и разработка моделей и методики представления и организации знаний с применением онтологического подхода и инструментальных средств Smart-технологии, при реализации образовательных программ и процессов» по грантовому финансированию Министерства образования и науки Республики Казахстан, № AP05134973, руководитель проекта к.т.н., доцент РГП на ПХВ "Институт информационных и вычислительных технологий" КН МОН РК, Кубеков Б.С.

**Структура и объем диссертации.** Диссертация состоит из введения, пяти разделов, заключения, списка использованных источников из 75 наименований источников и приложений.

# **1 КОНЦЕПЦИЯ ПАРАДИГМЫ И ФОРМАЛИЗМЫ МОДЕЛИРОВАНИЯ ЗНАНИЙ**

В данном разделе излагаются принятые в наших исследованиях концепции инженерии знаний, теоретические основы формализации знаний, применение онтологического инжиниринга в представлении и организации знаний семантической сети предметной области; применение парадигм объектного и порождающего программирования, а также основных принципов инженерии предметной области, связанных с созданием абстрактных представлений.

Показаны языковые конструкции, созданного для этих целей языка спецификации знаний, и применение конструкций в формальных рамках анализа общности и изменчивости понятий предметной области.

Рассматриваются сценарии работы программного интерпретатора адаптивной образовательной среды, связанные с линейным сопоставлением элементов выражения знания с правилами языка, а также со сборкой онтологической и характеристической моделей и их последующей визуализацией в виде ориентированного графа.

## **1.1 Обзор информационных технологий в образовательной практике**

В журнале SkReview, Сколково, отмечено, что Семантические технологии Web интенсивно внедряются в сферу образования. Семантические технологии представляют значения с помощью онтологии и обеспечивают аргументацию, используя связи, правила, логику и условия, оговоренные в онтологии.» [26].

20 марта на правительственном часе Мажилиса парламента с докладом выступил министр образования и науки РК Сагадиев Е.К.[27]. В докладе были рассмотрены вопросы о создании единой базы образовательных учреждений и мобильного доступа к ресурсам, цифровизации и прозрачность информационных ресурсов, улучшение качества знаний обучающихся с использованием лучших мировых практик, интеграция баз данных и объединение в общее хранилище. И здесь первоочередной задачей является создание онтологии для структуризации знаний в образовательных учреждениях.

В результатах обзора исследователей ближнего и дальнего зарубежья в области инженерии знаний, были даны рекомендации по созданию единого информационно-образовательного ресурса, с целью формирования онтологий предметных областей профессионального международного сообщества. Широко распространены мультязычные ресурсы, которые позволяют способствовать эффективному взаимодействию и согласованию терминологии на разных языках.

Обзор статистических данных подтверждает, что первые авторские работы на русском языке появились в 90-х, их количество заметно увеличилось с начала 2000-х, а в последние годы использование онтологий в образовательных целях является одной из обсуждаемых и перспективных областей искусственного интеллекта. Подтверждением служит следующая

статистические данные. На запрос «онтологии в образовании» по базе электронной библиотеки диссертаций disserCat мы получили 9841 результатов, а информационная база eLibrary на аналогичный запрос предоставляет 25617 результатов. Однако на поисковый запрос с уточненной формулировкой «технологии семантического веба в образовании» дается гораздо меньше результатов 290 по базе disserCat, и 233 по информационной базе eLibrary.

Перспективность проводимых исследований подтверждает тот факт, что многие из них поддержаны грантами различных организаций, включая Министерство Образования и Науки Республики Казахстан, АО "Национальное Агентство по Технологическому Развитию" и др. Работы выполняются в основном экспертами в области информационных технологий казахстанских НИИ, кафедр и научно-исследовательских лабораторий технических университетов. Также важно отметить рост числа ежегодных научных мероприятий по рассматриваемой тематике. В частности, в списке на 2017 год, приводятся такие конференции, как Международная научно-техническая конференция «Искусственный интеллект в инфокоммуникации», г. Алматы, Среди основных тем конференции- индустрия 4.0., искусственный интеллект, Big Data и машинное обучение; V Международная научно-практическая конференция «Интеллектуальные информационные и коммуникационные технологии – средство осуществления третьей индустриальной революции в свете Стратегии «Казахстан-2050», г. Астана, На конференции были озвучены вопросы международного обмена передовым опытом, планы реализации Стратегии «Казахстан-2050» в области создания, интеграции науки, образования и IT-индустрии и трансферта интеллектуальных информационных и коммуникационных технологий, в целях создания ИКТ-кластера. Также следует назвать конференцию «Инновационные технологии управления государством: Цифровой Казахстан», которая проводилась в Астане.

Тенденции роста интереса к онтологиям подтверждаются исследованиями Боргеста Н.М. «Границы онтологии проектирования», предложены статистические результаты материала международных научных журналов, таких как «Прикладные онтологии», «Онтология проектирования».

Созданием онтологических ресурсов и инструментов работы пользователей с ними занимаются такие научные центры, как Восточно-Казахстанский государственный технический университет им. Д. Серикбаева, к.т.н. Рахметуллина С.Ж. «Мониторинг развития вуза на основе онтологических моделей ситуаций», к.т.н. Уркумбаева А.М., Рохас Криулько Н.П. и Котлярова И.А. «Разработка онтологической базы знаний e-университета». Так же основным направлением научной деятельности в области создания онтологических ресурсов занимается передовой научный центр Республиканское государственное предприятие (РГП) на праве хозяйственного ведения (ПХВ) "Институт информационных и вычислительных технологий".

Обращаясь к тематике реализуемых образовательных проектов, можно выделить несколько направлений исследовательской деятельности:

1. администрирование учебного процесса в целом;
2. мобильная интеграция;
3. индивидуальный подход к обучению;
4. проектирование профессиональных компетенций специалистов;
5. автоматизация тестирования.

Требования МОН и международных стандартов согласуют рекомендации по использованию онтологий для организации администрирования учебного процесса с учетом моделей компетентностного подхода. В частности, из ученых ближнего зарубежья можно выделить С. Н. Дегтярева, которым предлагается управление учебно-познавательной деятельностью учащихся на основе онтологической модели [1]. Предпосылки к использованию прикладных задач на основе семиотико-онтологической модели обучения представлены в исследованиях О. М. Топорковой [2]. Использование онтологий в учебных процессах не ограничивается информационными порталами и информационно-образовательными ресурсами, наиболее перспективным направлением являются системы извлечения знаний из текстов. Л. Р. Черняховской и др. создана онтологическая модель проведен анализ систем электронного обучения, [3]. Описание методических аспектов процесса обучения и создание онтологической модели реализованы в исследованиях Уфимского государственного авиационного технического университета [28]. Разделение онтологий в порталах знаний по тематикам для повышения эффективности поиска предложены О. И. Боровиковым [29]. Оценка релевантности, индексирование и поиск ресурсов предложены в исследованиях Загорулько Е.А. Исследования И. И. Мирошниченко и И. Ю. Шполянской основаны на онтологической модели учебного подразделения вуза [4].

Описание методов по автоматическому извлечению знаний изложены в трудах А. Е. Ермакова, А. В. Маликова и А. С. Целиковского [6,30,31]. Дидактика семантические карты исследованы в работах С.В. Титенко, О. Е. Дороховой, Загорулько Ю.А. др ученых, предложены методы визуализации дидактических онтологий [5,32,33].

Необходимо отметить активность ученых Казахстана, в исследованиях А. Ж. Кинтонова, М. Б. Рахижанова и Шарипбаева А. А. рассматривалась модель многоуровневой подготовки специалистов (рисунок Б.3) на основе технологий семантического веба [34]. По мнению А. Р. Давлетбаевой С. И. Родзина и Л. С. Родзиной предлагается циклическая модель управлением качества учебного процесса [28,35,36].

В рамках представленных научных исследованиях существуют предметно-ориентированные онтологии, такие как «Автомобили» [6]; «Общая физика» [37]; «Онтологии-схемы» [38]; «Математическая логика [39]; «Электронные образовательные ресурсы» [40] и др. Исследования зарубежных ученых ведут к автоматизации процесса тестирования обучающихся и автоматизированного создания контрольных вопросов на основе метапредметных понятий учебного материала [41,42].



Одним из распространённых редакторов, в предложенном обзоре исследований, по работе с онтологиями является Protégé [43 и др.]. Расширение редактора возможно с использованием плагинов, таких как TGVizTab и Jambalaya [44]. Возможно использование среды Eclipse 3.4 на языке Java [45].

Представленный краткий обзор не только отражает суть исследования, но и помогает раскрыть основные возможности онтологий в системах образования, а также указывает на отдельные особенности существующих методов и моделей последующей обработки онтологий.

Образовательные технологии - неотъемлемая часть содержания обучения, этот термин, обозначает формы передачи и генерирования знаний на основе технической базы. При ознакомлении и анализе методик и технологий, применяемых в образовании, возникают различные ассоциации используемых понятий и моделей с понятиями в информационных технологиях и в, частности, с понятиями, связанными с парадигмами объектно-ориентированного, порождающего программирования и мультипарадигменного проектирования.

Необходимость использования концепций, механизмов и инструментальных средств информационных технологий в образовательной практике и технологиях уже не вызывает споров, и в этом смысле наиболее приемлемой формой связи мышления между этими технологиями является аналогическое мышление, выражением которого является метафорическое мышление [46]. Например, введенное нами понятие - визуальная понятийная модель - это метафора, позволяющая соединить в себе такие понятия объектной технологии, как класс, наследование, отношения композиции и агрегации с сущностями учебного контента дисциплин обучения вместе с ассоциациями, совместно определяющими как структуру учебного контента, так и механизмы взаимодействия сущностей в различных сценариях планирования, разработки и обучения.

Образовательные технологии базируются на современных достижениях информационных технологий, на привлечении инженерии знаний и, в частности, онтологического инжиниринга. Несмотря на разнообразные методы и средства, потребуются существенные усилия для создания методик и методов представления и организации знаний, а также для проектирования образовательных ресурсов, которые позволяли бы в полной мере использовать множество педагогических, образовательных и информационных парадигм в рамках единой образовательной парадигмы.

Парадигмы образования [47] - это некие руководства для работников в сфере образования при организации процесса обучения или управления образованием. Каждая образовательная парадигма реализуется через свои цели, принципы, содержание и свои соответствующие технологии. Смена парадигм связана с изменением мышления, мировоззрения у большинства работников образования[53].

Переход на новую парадигму образования требует от высшей школы не просто репродуктивной передачи данных от преподавателя к студенту, а формирование активной учебной деятельности[54].

Парадигма - термин, широко используемый в современном проектировании систем и обозначающий способ построения системных абстракций на основе свойств общности и изменчивости.

Под парадигмой образования будем понимать базовую модель конкретного способа организации учебной информации, на основе свойств общности и изменчивости и, в целом, как ведущий подход к выбору содержания и форм организации образовательных ресурсов.

Рассмотрим возможность представления образовательных ресурсов с помощью онтологии и характеристической модели [55,56].

Онтология определяет концептуализацию, которая лежит в основе формализма представления знания. Моделирование характеристик является основной методикой выявления и фиксации общности и изменчивости на понятиях в онтологии и на свойствах характеристик, что дает возможность разрабатывать повторно используемые образовательные компоненты и использовать их как при проектировании дисциплин учебного плана специальности, так и для адаптации образовательных программ к требуемым профессиональным компетенциям выпускника.

Образовательные системы должны быть гибкими и адаптивными за счет анализа и реализации абстракций общности и изменчивости образовательных средств и ресурсов, что накладывает определенную обязанность по подбору таких абстракций, которые со временем будут оставаться неизменными и адаптируемыми к новым условиям и требованиям [66-71]. Такая стабильность наиболее характерна для абстракций общеобразовательных дисциплин учебного плана, частично для базовых дисциплин, и в большей степени изменчивость характерна для абстракций цикла профилирующих дисциплин специальности. Поэтому при планировании учебного процесса следует основываться прежде всего на профилирующих дисциплинах и на знании о том, каким изменениям будут подвергаться требования к профессиональным компетенциям выпускника в будущем.

Применение онтологии и характеристических моделей для обучения требует: во-первых, проведение анализа структуры и организации контента учебного материала дисциплин; во-вторых, создание визуально-мыслительных образов, с помощью ассоциативного связывания понятий и структурных элементов содержания учебного материала для формирования целостной системы знаний в рамках отдельной дисциплины, курса и специальности; и в-третьих, включение механизмов взаимодействия визуально-мыслительных образов учебного материала, способствующих приобретению обучающимися профессиональных навыков и практических умений.

Процесс построения онтологии начинается с определения базовых абстракций предметной области, так называемых, опорных понятий, составляющих семантические знания предметной области. Каждое опорное понятие, в свою очередь, может быть идентифицировано набором конкретизирующих понятий. Каждое такое представление, таким образом, есть онтология опорного понятия, представляемая в виде понятийного графа.

Если онтология используется для представления и организации знаний предметной области, то с помощью характеристической модели можно описывать свойства общности и изменчивости понятий онтологии (Рисунок Б.5).

Моделирование характеристик - самая значимая заслуга инженерии знаний [58,64,65]. Проектирование систем обучения невозможно без образовательных компонент повторного использования, что, в свою очередь, невозможно без анализа и представления общности и изменчивости понятий дисциплин специальности и соответствующих образовательных средств и учебных ресурсов [59,63]. Образовательные средства и учебные ресурсы многократного применения должны отличаться от традиционных учебных ресурсов значительно более существенной изменчивостью, а основной методикой выявления и фиксации изменчивости является именно моделирование характеристик. Характеристики совершенно незаменимы при формулировании кратких описаний учебных ресурсов, между экземплярами которых существуют значительные различия, а также в дифференциации экземпляров этих ресурсов.

Таким образом, характеристические модели дают возможность применения дополнительных формализмов для моделирования и представления семантического содержания образовательных средств и учебных ресурсов.

Если онтология используется для представления и организации знаний предметной области, то с помощью характеристической модели описываются свойства общности и изменчивости понятий онтологии.

Процесс построения онтологии начинается с определения базовых абстракций предметной области - опорных понятий, составляющих семантические знания предметной области, каждое из которых идентифицируется набором конкретизирующих понятий. Визуально онтология опорного понятия представляется понятийным графом.

По мере развития базы знаний содержания учебных дисциплин следует разработать общий подход по онтологическому анализу учебного контента и автоматизации процессов представления, организации и визуализации знаний на базе повторно используемых образовательных компонентов и, как следствие, созданию интегрированных учебных курсов, совместных программ обучения в рамках реализации академической мобильности и возможности гибкого планирования учебного процесса [60,61].

## **1.2 Анализ существующих формализмов моделирования понятий и адаптация к требованиям**

Формализация - метод исследования, в основе которого лежит отображение содержательного знания в знаково - символическом виде (формализованном языке). К наиболее часто используемым формам отображения знаний традиционно относятся продукционная, сетевая, фреймовая, алгебраическая модели, графы и множества. В этом ряду, на наш взгляд, приемлемой формой отображения педагогических моделей и

образовательных ресурсов, подходит объектно-ориентированная парадигма, которая располагает большим набором возможностей для моделирования, конструирования и визуализации образовательных ресурсов, а также средствами моделирования свойств и механизмов их взаимодействия [62].

В искусственном интеллекте знания о предметной области представляются в виде иерархии структурированных объектов, связанных между собой отношениями. На этой идее базируются такие формализмы отображения знаний, как фреймы, семантические сети, а также язык объектного моделирования UML, который являясь языком представления знаний в виде иерархии структурированных классов, позволяет описывать декларативные знания предметной области.

Декларативные знания - это знания о мире задачи, то есть знания, описывающие свойства предметной области, в которой решается поставленная задача. К сожалению, перечисленные формализмы лишены формальной семантики, поскольку выражаемая в них информация предназначена для человеческого, а не для машинного восприятия. Такое описание можно было бы сделать с использованием логики предикатов первого порядка. Но с ней существует ряд проблем: нет удобной поддержки иерархий, нет удобных средств описания структурированных классов, а также логический вывод разрешим наполовину.

Таким образом, для представления знаний в базах знаний параллельно развивались и использовались такие формализмы как:

- семантические сети и фреймы, где было все, кроме формальной семантики;
- логика предикатов, в которой была семантика, но не было удобных средств для организации знаний.

В основу дальнейших построений приняты концепции:

- инженерии и формализации знаний;
- онтологического инжиниринга в представлении и организации семантических знаний учебного контента;
- объектного и порождающего программирования;
- построения системных абстракций учебного контента на основе свойств общности и изменчивости.

Под парадигмой образования будем понимать базовую модель конкретного способа организации учебной информации, на основе свойств общности и изменчивости и, в целом, как ведущий подход к выбору содержания, форм организации образования и способов обучения.

Для спецификации знаний в формате онтологии используются отношения: "Композиция", "Агрегация" и "Альтернативный выбор".

Вводимые далее правила представления знаний основаны на следующих концепциях:

1) Семантическое знание образовательного ресурса можно представить произвольным набором опорных и только опорных понятий учебного контента, каждое из которых подлежит идентификации своими дочерними понятиями.

2) Понятием (concept) назовем любую мысль, в которой отражаются общие существенные свойства и связи предметов и явлений учебного контента. Понятия можно представить как "ориентиры", с помощью которых происходит систематизация знаний. В то же время, понятия субъективны, так как их семантика определяется контекстом применения.

3) Онтология - это подробная спецификация понятийной (терминологической) структуры учебного контента.

Основное назначение онтологии заключается в определении формальной семантики знания в сочетании с удобными формами хранения и представления для восприятия человеком.

Потребность в разработке онтологии возникает в связи с необходимостью:

- совместно использовать общее понимание структуры информации;
- повторно использовать знания в предметной области;
- отделять знания в предметной области от оперативных знаний;
- проводить анализ знаний в предметной области.

4) Онтология определяется как тройка множеств:  $Om = \langle C, R, F \rangle$ , где  $C$  - множество понятий (терминов) учебного контента;  $R$  - множество отношений между понятиями;  $F$  - множество функций интерпретации, определения которых заданы на отношениях между понятиями в онтологии.

5) Родительское понятие - как продукт абстрагирования, есть абстрактный компонент общего характера, выражающий общность для всех своих дочерних понятий. В дальнейших построениях под родительским понятием будем понимать опорное понятие учебного контента.

6) Экземпляром родительского понятия будем считать конечное множество конкретизирующих понятий онтологии, связанных друг с другом отношениями "Композиция", "Агрегация" и "Альтернативный выбор", с помощью которых осуществляется семантическая идентичность каждого из конкретизирующих понятий с его родительским понятием.

7) Визуально онтологию можно представить ориентированным графом -  $G$ , вершинами которого являются понятия, а ребрами - отношения между ними. Корневая вершина графа - опорное понятие учебного контента, идентификация которого ограничивается не более чем двумя уровнями дочерних понятий иерархии, в которых любое необязательное конкретизирующее понятие является висячей вершиной, то есть вершиной, не имеющей своих дочерних конкретизирующих понятий.

8) Для систематизации конкретизирующих понятий используем категорию "Семантическая идентичность", для которой введем следующие определения функции интерпретации:

Определение 1. Отношение "Композиция" - отношение, в котором отражается свойство общности для дочернего понятия, и обязательность присутствия дочернего понятия во всех экземплярах родительского понятия.

Определение 2. Отношение "Агрегация" - отношение, в котором отражается свойство общности для дочернего понятия, и необязательность присутствия дочернего понятия в экземплярах родительского понятия.

Определение 3. Отношение "Альтернативный выбор" - отношение, в котором отражается свойство изменчивости дочернего понятия и необязательность его присутствия в экземплярах родительского понятия.

Для идентификации и спецификации перечисленных отношений над понятиями онтологии было введено так называемое выражения знания – специально разработанная нотация на основе алгебраической модели отображения знаний.

1) Семантика идентификации и спецификация отношения "Композиция"

Считаем, что отношение "Композиция" порождает набор обязательных конкретизирующих дочерних понятий родительского понятия, при условии, что родительское понятие является обязательным понятием учебного контента.

Данное отношение предполагает использование обязательных конкретизирующих понятий только в рамках данного родительского понятия, придавая им некоторую уникальность. Аналогичное отношение используется в языке объектно-ориентированного моделирования UML с таким же названием.

Если родительское понятие является необязательным, тогда оно не может быть подвергнуто дальнейшей конкретизации, что справедливо и для любого конкретизирующего понятия.

Пусть  $C_i$  - опорное понятие предметной области, тогда  $C_i$ , как композиция конкретизирующих понятий, может быть отображена следующим выражением знания:

$$C_i \leq *C_{i1} *C_{i2} \dots *C_{ik}; \quad (1.1)$$

где символ '\*' обозначает обязательность использования конкретизирующего понятия для идентификации опорного понятия  $C_i$ .

В выражении знания символ ' $\leq$ ' обозначает отношение «Импликация» - это отношение, при котором набор конкретизирующих понятий  $C_{i1}, C_{i2}, \dots, C_{ik}$  влечет идентификацию опорного понятия  $C_i$ . Импликация, таким образом, ассоциируется с причинностью.

2) Семантика идентификации и спецификация отношения "Агрегация"

Отношение "Агрегация" - это сочетание обязательных и необязательных дочерних понятий родительского понятия, при условии, что родительское понятие является обязательным понятием учебного контента.

Данное отношение предполагает использование обязательных конкретизирующих понятий не только в рамках данного родительского понятия. Аналогичное отношение используется в языке объектно-ориентированного моделирования UML с таким же названием.

Если родительское понятие является необязательным, тогда оно не может быть подвергнуто дальнейшей конкретизации, что справедливо и для любого конкретизирующего понятия.

Пусть  $C_i$  - опорное понятие учебного контента, тогда агрегация конкретизирующих понятий, может быть отображена следующим выражением знания:

$$C_i \leq *C_{i1} + C_{i2} \dots *C_{ik}; \quad (1.2)$$

где символ '+' обозначает необязательность конкретизирующего понятия для идентификации опорного понятия  $C_i$ , а обязательные понятия  $C_{i1}$  и  $C_{ik}$  могут быть использованы не только в рамках данного родительского понятия.

3) Семантика идентификации и спецификация отношения "Альтернативный выбор"

Отношение "Альтернативный выбор" - определено для различных сочетаний обязательных, либо различных сочетаний обязательных и необязательных понятий родительского понятия, при условии, что родительское понятие является обязательным понятием учебного контента.

В отношении «Альтернативный выбор», также как и в отношении «Агрегация», предполагается использование обязательных конкретизирующих понятий не только в рамках данного родительского понятия.

Пусть  $C_i$  - опорное понятие предметной области, тогда идентификацию  $C_i$  через отношение «Альтернативный выбор», можно отобразить одним из следующих допустимых выражений знания:

$$C_i \leq *C_{i1} \sim + C_{i2}; \quad (1.3)$$

где символ '~' - тильда, обозначает возможность альтернативного выбора обязательного  $C_{i1}$ , либо необязательного конкретизирующего понятия  $C_{i2}$ , или

$$C_i \leq *C_{i1} \sim * C_{i2}; \quad (1.4)$$

как возможность альтернативного выбора одного из обязательных понятий -  $C_{i1}$ ,  $C_{i2}$ .

Приведем примеры возможных случаев идентификации опорного понятия  $C_i$  следующими выражениями знания:

1)  $C_i \leq *C_{i1} * C_{i2} * C_{i3}$ ; в данном выражении знания опорное понятие  $C_i$  - идентифицируется композицией из трех обязательных дочерних понятий. В этом выражении знания обязательные конкретизирующие понятия должны использоваться только в рамках данного опорного понятия.

2)  $C_i \leq *C_{i1} * C_{i2} + C_{i3}$ ; в данном выражении знания опорное понятие  $C_i$  - идентифицируется композицией из двух обязательных и агрегацией одного необязательного дочерних понятий. В этом выражении знания обязательные конкретизирующие понятия могут использоваться не только в рамках данного опорного понятия.

3)  $C_i \leq *C_{i1} * C_{i2} \sim + C_{i3}$ ; в данном выражении знания опорное понятие  $C_i$  - идентифицируется композицией из двух обязательных и агрегацией одного необязательного дочерних понятий, причем для обязательного понятия  $C_{i2}$  и необязательного понятия  $C_{i3}$  существует альтернатива выбора. В этом

выражении знания обязательные конкретизирующие понятия могут использоваться не только в рамках данного опорного понятия.

4)  $C_i \leq *C_{i1} + C_{i2} \sim + C_{i3}$ ; в данном выражении знания опорное понятие  $C_i$  идентифицируется обязательным понятием  $C_{i1}$  и агрегацией из двух необязательных дочерних понятий  $C_{i2}$  и  $C_{i3}$ , для последних которых существует альтернатива выбора. В этом выражении знания обязательное конкретизирующее понятие может использоваться не только в рамках данного опорного понятия.

5)  $C_i \leq *C_{i1} * C_{i2} \sim * C_{i3}$ ; в данном выражении знания опорное понятие  $C_i$  идентифицируется композицией из трех обязательных дочерних понятий, причем для обязательных понятий  $C_{i2}$  и  $C_{i3}$  существует альтернатива выбора. В этом выражении знания обязательные конкретизирующие понятия могут использоваться не только в рамках данного опорного понятия.

6)  $C_i \leq *C_{i1} (*C_1 + C_2) * C_{i2}$ ; в данном выражении знания опорное понятие  $C_i$  идентифицируется композицией из двух обязательных  $C_{i1}$  и  $C_{i2}$  дочерних понятий, где обязательное понятие  $C_{i1}$  конкретизируется набором из обязательного дочернего понятия  $C_1$  и необязательного дочернего понятия  $C_2$ , семантическая трактовка которых определена в контексте обязательного понятия  $C_{i1}$ . В этом выражении знания обязательные конкретизирующие понятия могут использоваться не только в рамках данного опорного понятия.

7)  $C_i \leq *C_{i1} (*C_1 \sim + C_2) + C_{i3}$ ; в данном выражении знания опорное понятие  $C_i$  идентифицируется композицией из обязательного  $C_{i1}$  и агрегацией необязательного  $C_{i3}$  дочерних понятий, причем обязательное понятие  $C_{i1}$  имеет свой уровень конкретизации, как альтернатива выбора обязательного  $C_1$ , либо необязательного понятия  $C_2$ , семантическая трактовка которых определена в контексте понятия  $C_{i1}$ . В этом выражении знания обязательные конкретизирующие понятия могут использоваться не только в рамках данного опорного понятия.

Таким образом, идентификация опорного понятия  $C_i$  представляется иерархической структурой понятий из двух уровней, отношения между понятиями которой позволяют:

- идентифицировать семантические знания учебного контента наборами опорных понятий;
- идентифицировать опорное понятие иерархической структурой из конкретизирующих понятий, наделенных свойствами общности и изменчивости;
- представлять характеристические свойства описания экземпляров опорных понятий с помощью характеристической модели;
- конструировать онтологии для моделирования опорных понятий учебного контента образовательной дисциплины.

На рисунке 1.1 показан пример онтологии опорного понятия  $C_i$ , идентифицируемого понятиями первого уровня -  $C_{i1}$ ,  $C_{i2}$ ,  $C_{i3}$  и  $C_{i4}$ , из которых  $C_{i2}$  определяет семантический контекст понятий второго уровня -  $C_1, C_2$ .



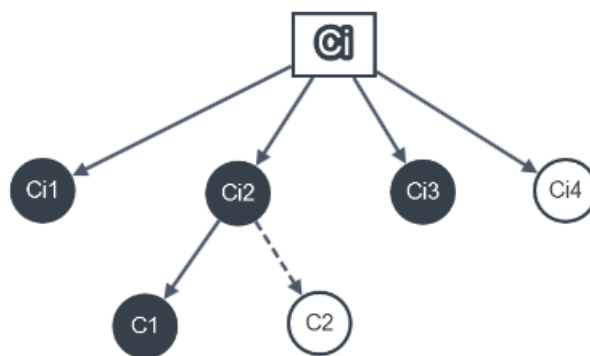


Рисунок 1.1 - Понятийный граф онтологии опорного понятия  $C_i$

Выражение знания опорного понятия  $C_i$  имеет вид:

$$C_i \leq *C_{i1} * C_{i2} (*C_1 \sim + C_2) * C_{i3} + C_{i4}; \quad (1.5)$$

где символ '~' - тильда в выражении знания определяет отношение "альтернативный выбор" для конкретизирующих понятий второго уровня  $C_1$  и  $C_2$ .

Для построения понятийных графов применяется среда Cytoscape.js, в которой обязательное понятие изображается закрашенным кругом, необязательное - не закрашенным, а отношение "альтернативный выбор" - пунктирной стрелкой.

### 1.3 Инженерия образовательной предметной области

Методика, основанная на инженерии образовательной предметной области, имеет целью осуществлять реализацию учебных планов и программ на основе семейства знаниевых компонентов. Общность требований к отдельным компонентам в рамках одной образовательной предметной области обуславливает схожесть их характеристик. Следовательно, знания, накопленные преподавателем, разработавшим курсы и учебно-методические комплексы дисциплин, могут в дальнейшем помочь ему при подготовке новых курсов, в том числе и в смежных направлениях. Накопление подобных знаний образовательной предметной области в виде знаниевых компонент многократного применения и их повторное использование при разработке новых курсов поможет гибко формировать необходимые образовательные тренды и соответствующие учебные программы.

Можно привести пример курсов, связанных с операционными системами и компьютерными сетями, знаниевые компоненты которых, с появлением облачных технологий, составили базовый знаниевый контекст курса "Облачные вычисления".

### 1.4 Формализмы моделирования характеристик

Характеристика - различимое существенное свойство, а также значимый и доступный аспект понятия. Характеристика, с точки зрения концептуального

моделирования, представляет собой важное свойство, позволяющее обозначать сходства и различия между экземплярами понятия.

Характеристики незаменимы при формулировании краткого описания свойств понятия и для идентификации различий между экземплярами понятия.

Под характеристической моделью  $F_m$  будем понимать модель с обобщением характеристик и изменяемых параметров понятия, а также логическим обоснованием композиции и агрегации характеристик для спецификации свойств понятия и его экземпляров. В характеристической модели фиксируются все свойства, которые касаются возможности конфигурации компонент многократного применения и их повторного использования.

Спецификация понятия осуществляется с помощью набора характеристик (feature), отношения между которыми подчинены формализмам и правилам, введенным выше для онтологии.

Моделирование характеристик - это непрерывный итеративный процесс, состоящий из следующих этапов:

- анализ общности и различия на свойствах понятия и идентификация его характеристик;
- структуризация и систематизация характеристик понятия в рамках онтологии и их классификация на обязательные, необязательные и альтернативные;
- анализ допустимых сочетаний характеристик и взаимосвязей между ними для спецификации понятия, и при определении механизмов их использования для поиска новых характеристик;
- сбор и анализ дополнительных сведений, связанных с характеристиками, например, для семантического описания, логического обоснования характеристик, ограничений, правил зависимостей между характеристиками, а также открытых и закрытых свойств и приоритетов характеристик.

На рисунке 1.2 показана характеристическая модель  $F_m$  понятия  $C_i$ .

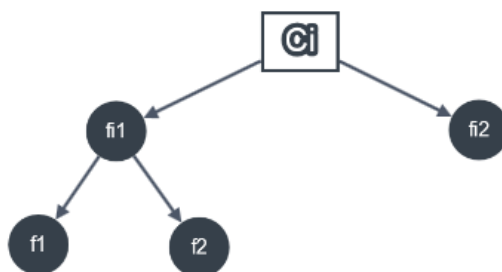


Рисунок 1.2 - Характеристическая модель понятия  $C_i$ .

Здесь мы видим, что понятие  $C_i$  есть композиция обязательных конкретизирующих характеристик  $f_{i1}$  и  $f_{i2}$ , причем характеристика  $f_{i1}$  имеет свой уровень конкретизации, как композиция дочерних обязательных характеристик

$f_1$  и  $f_2$ . Таким образом, экземплярами понятия  $C_i$  могут быть допустимые наборы из следующих обязательных характеристик:  $\{f_{i1}, f_1, f_2, f_{i2}\}$ .

Спецификацию опорного понятия  $C_i$  характеристической моделью назовем выражением спецификации, которое будет иметь следующий вид:  $C_i \leq *f_{i1}(*f_1*f_2)*f_{i2}$  ;

Свойство изменчивости понятия может моделироваться сочетанием обязательных и необязательных характеристик, либо как альтернатива между обязательными характеристиками, либо как альтернатива между обязательными и необязательными характеристиками, как изображено на рисунке 1.3.

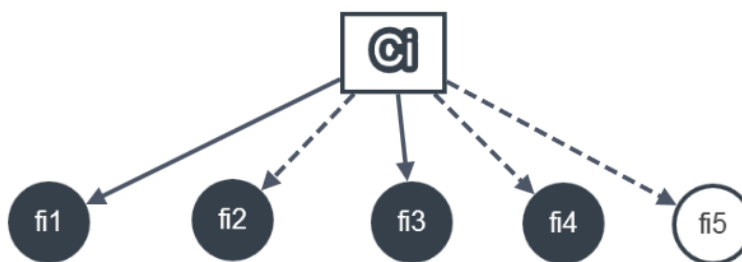


Рисунок 1.3 - Моделирование свойства изменчивости понятия с помощью характеристической модели

Приведенная на рисунке 1.3 идентификация опорного понятия  $C_i$  характеристической моделью будет отождествляться следующим выражением спецификации:

$$C_i \leq *f_{i1} \sim *f_{i2} *f_{i3} \sim *f_{i4} \sim +f_{i5}; \quad (1.6)$$

В приведенном выражении спецификации у понятия  $C_i$  есть два набора обязательных характеристик с альтернативным выбором - набор из характеристик  $f_{i1}$  и  $f_{i2}$ , и набор из обязательных характеристик  $f_{i3}$ ,  $f_{i4}$  и необязательной характеристикой  $f_{i5}$  с альтернативным выбором.

В этом случае возможны следующие варианты экземпляров опорного понятия  $C_i$ :  $\{C_i, f_{i1}, f_{i3}\}$ ,  $\{C_i, f_{i1}, f_{i4}\}$ ,  $\{C_i, f_{i1}, f_{i5}\}$ ,  $\{C_i, f_{i2}, f_{i3}\}$ ,  $\{C_i, f_{i2}, f_{i4}\}$  или  $\{C_i, f_{i2}, f_{i5}\}$ .

Рассмотрим интерпретацию введенных правил, отношений и структур представления знаний для образовательной предметной области, на примере некоторой дисциплины  $D$ .

Представим  $O_m$ - онтологию произвольного опорного понятия дисциплины, характеристическую модель  $F_m$  опорного понятия и выражение знания соответственно.

Введем следующие обозначения:

1.  $t(D) = (C_1, \dots, C_i, \dots, C_n)$  - опорные понятия учебного контента  $t$ -й темы дисциплины  $D$ ;

2.  $O_m$  - онтология опорного понятия учебного контента  $t$  -й темы дисциплины  $D$ ;

3.  $KE$  - выражение знания опорного понятия учебного контента  $t$  -й темы дисциплины  $D$ ;

4.  $F_m$  - характеристическая модель опорного понятия учебного контента  $t$  -й темы дисциплины  $D$ .

Контекст - (от лат. contextus - сцепление, соединение, связь) - относительно законченный по смыслу отрывок текста, в пределах которого наиболее точно и конкретно выявляется смысл и значение отдельно входящего в него слова, фразы, совокупности фраз.

В наших построениях под контекстом:

- $m$ -го опорного понятия будем понимать  $t$ -ю тему дисциплины;
- $t$ -й темы будем понимать дисциплину  $D$ , в которую входит данная тема;
- онтологии, выражения знания и характеристической модели будем понимать опорное понятие  $t$ -й темы дисциплины  $D$ .

Другими словами, под "контекстом" понимается некоторое понятийное информационное пространство, в пределах которого любое понятие наделяется семантической общностью и индивидуальными свойствами, присущими данному контексту.

Далее рассмотрим следующие операции онтологического инжиниринга:

- организация онтологической модели опорного понятия и спецификация онтологической модели выражением знания;
- организация характеристической модели опорного понятия и спецификация характеристической модели выражением спецификации.

В качестве первого примера применения введенных формализмов для организации онтологии и спецификации выражения знания понятия, а также для организации характеристической модели и ее выражения спецификации, рассмотрим тему: "Указатель в языке программирования Си" дисциплины "Технология программирования", читаемой на первом курсе студентам специальности "Вычислительная техника и программное обеспечение".

В данной теме выделим опорные понятия, представим их онтологии, идентификацию опорных понятий и рассмотрим характеристические модели понятий.

Рассмотрим опорное понятие темы  $C$  - "Указатель на объект программы". Идентифицирующими понятиями для  $C$  являются:  $C_1$  - "Указатель на статический объект программы" и  $C_2$  - "Указатель на динамический объект программы", что можно представить выражением знания:  $C \leq +C_1 * C_2$ ; состоящим из необязательного понятия  $C_1$  и обязательного понятия  $C_2$ .

Далее, представим характеристическую модель опорного понятия "Указатель на объект программы", включающую следующие характеристики:

- $f_1$  - значение переменной указательного типа,
- $f_2$  - прямой доступ к объекту программы,
- $f_3$  - косвенный доступ к объекту программы,

–  $f_4$  - операции над указателями.

Таким образом, характеристическую модель опорного понятия - "Указатель на объект программы" можно представить следующим выражением спецификации:

$$C \leq *f_1 + f_2 * f_3 * f_4; \quad (1.7)$$

как композицию обязательных конкретизирующих характеристик  $f_1$ ,  $f_3$ ,  $f_4$ , и агрегацию необязательной характеристики  $f_2$ .

На рисунке 1.4 показан характеристический граф опорного понятия  $C$  - "Указатель на объект программы":

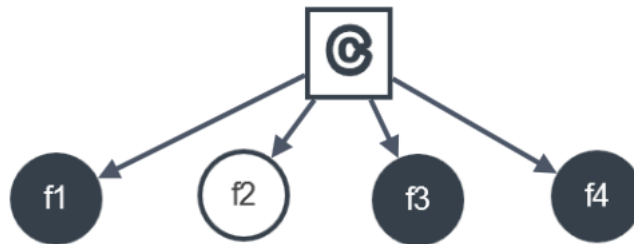


Рисунок 1.4 - Характеристический граф опорного понятия  $C$

Для конкретизирующего понятия  $C_1$  - "Указатель на статический объект программы" можно выделить следующие характеристики:  $f_{11}$  - унарный оператор '&',  $f_{12}$  - унарный оператор '\*',  $f_{13}$  - указатель на константу и  $f_{14}$  - константный указатель. Тогда характеристическую модель конкретизирующего понятия  $C_1$  - "Указатель на статический объект программы", можно представить следующим выражением спецификации:

$$C_1 \leq *f_{11} * f_{12} + f_{13} + f_{14}; \quad (1.8)$$

как композицию обязательных  $f_{11}$ ,  $f_{12}$  и агрегацию необязательных дочерних характеристик  $f_{13}$  и  $f_{14}$ .

На рисунке 1.5 представлен характеристический граф конкретизирующего понятия  $C_1$  - "Указатель на статический объект программы":

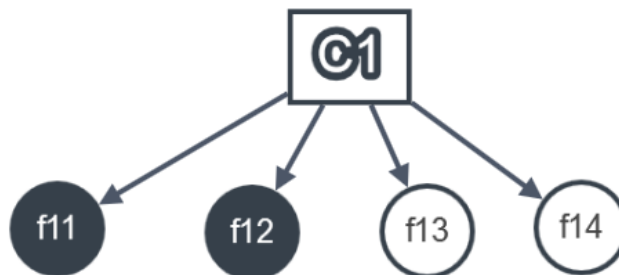


Рисунок 1.5 - Характеристический граф конкретизирующего понятия  $C_1$

Для конкретизирующего понятия  $C_2$  - "Указатель на динамический объект программы" выделим следующие характеристики:

- $f_{21}$  - динамическое выделение памяти,
- $f_{22}$  - динамическое освобождение памяти,
- $f_{23}$  - доступ к динамическому объекту по синтаксису выражения с указателем по смещению,
- $f_{24}$  - доступ к объекту с помощью операции с индексацией,
- $f_{25}$  - косвенное обращение к памяти через многоуровневый указатель.

Тогда характеристическую модель конкретизирующего понятия  $C_2$  - "Указатель на динамический объект программы" можно представить следующим выражением спецификации:

$$C_2 \leq *f_{21}*f_{22}*f_{23}+f_{24}*f_{25}; \quad (1.9)$$

и соответствующим графом, представленным на рисунке 1.6:

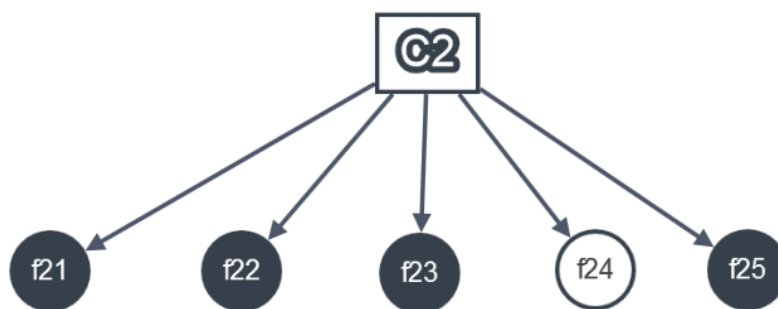


Рисунок 1.6 - Характеристический граф конкретизирующего понятия  $C_2$

Во всех приведенных примерах идентификации опорного понятия - "Указатель на объект программы" достаточным оказался один уровень конкретизации.

В качестве второго примера рассмотрим область знания программной инженерии SWEBOOK - «Требования к программному обеспечению».

Опорные понятия этой области знания составляют структурные элементы онтологии, являющиеся обязательными понятиями по определению.

Семантические знания области знания "Требования к программному обеспечению", представим набором из семи опорных понятий:  $C_1, C_2, C_3, C_4, C_5, C_6, C_7$  и их идентифицирующими понятиями:

- 1)  $C_1$  - Бизнес - требования:
  - \* $C_{11}$  - бизнес-правила,
  - \* $C_{12}$  - ограничения на поведение системы,
  - \* $C_{13}$  - документ - концепция.
- 2)  $C_2$  - Требования пользователя:
  - \* $C_{21}$  - функциональные требования пользователя,

\*C<sub>22</sub> - нефункциональные требования пользователя,

\*C<sub>23</sub> - сценарий событий. Данное понятие является контекстом второго уровня конкретизации, включающего следующие дочерние понятия: \*C<sub>1</sub>- вариант использования; +C<sub>2</sub> - диаграмма потоков данных; либо +C<sub>3</sub> - диаграмма переходов состояний; либо +C<sub>4</sub> - таблица "событие-отклик".

3) C<sub>3</sub> - Требования предметной области:

\*C<sub>31</sub> - функциональные требования;

\*C<sub>32</sub> - нефункциональные требования. Данное понятие является контекстом второго уровня конкретизации, включающего следующие дочерние понятия: \*C<sub>1</sub>- цели продукта; \*C<sub>2</sub> - атрибуты качества; +C<sub>3</sub> - ограничения внешнего вида и структуры продукта.

4) C<sub>4</sub> - Системные требования:

\*C<sub>41</sub> - организация системных требований,

\*C<sub>42</sub> - метрики качества системных требований,

+C<sub>43</sub> - обратные требования.

5) C<sub>5</sub> - Нефункциональные требования:

\*C<sub>51</sub> - требования к продукту. Данное понятие является контекстом второго уровня конкретизации, включающего следующие дочерние понятия: \*C<sub>1</sub> - требования к эксплуатации; \*C<sub>2</sub> - требования к эффективности; \*C<sub>3</sub> - требования к надежности; +C<sub>4</sub> - требования к переносимости; \*C<sub>5</sub> - обработка ошибок; \*C<sub>6</sub> - интерфейсные требования.

\*C<sub>52</sub> - организационные требования. Данное понятие является контекстом второго уровня конкретизации, включающего следующие дочерние понятия: \*C<sub>1</sub> - выходные требования; +C<sub>2</sub> - требования на реализацию; \*C<sub>3</sub> - требования на стандарты.

\*C<sub>53</sub> - внешние требования. Данное понятие является контекстом второго уровня конкретизации, включающего следующие дочерние понятия: \*C<sub>1</sub> - требования на взаимодействие; +C<sub>2</sub>- этические требования; +C<sub>3</sub> - юридические требования.

б) C<sub>6</sub> - Аттестация требований:

\*C<sub>61</sub>- достоверность,

+C<sub>62</sub>- непротиворечивость,

\*C<sub>63</sub>- полнота,

\*C<sub>64</sub>- выполнимость.

7) C<sub>7</sub> - Управление требованиями:

\*C<sub>71</sub>- планирование управления,

\*C<sub>72</sub> - управление изменениями.

Для опорных понятий построим выражения знаний и визуально отобразим каждое из них понятийным графом.

Выражением знания для опорного понятия C<sub>1</sub> - Бизнес - требования, будет выражение:

$$C_1 \leq *C_{11} * C_{12} * C_{13}; \quad (1.10)$$

а понятийный граф будет выглядеть следующим образом, как представлено на рисунке 1.7:

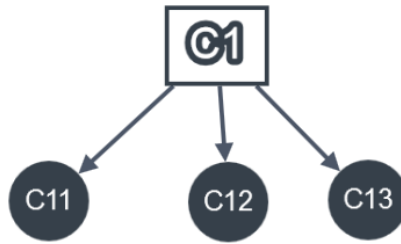


Рисунок 1.7 - Понятийный граф опорного понятия  $C_1$

Выражением знания для опорного понятия  $C_2$  - Требования пользователя, будет выражение:

$$C_2 \leq *C_{21} * C_{22} * C_{23} (*C_1 + C_2 \sim + C_3 \sim + C_4); \quad (1.11)$$

а понятийный граф будет выглядеть следующим образом как представлено на рисунке 1.8:

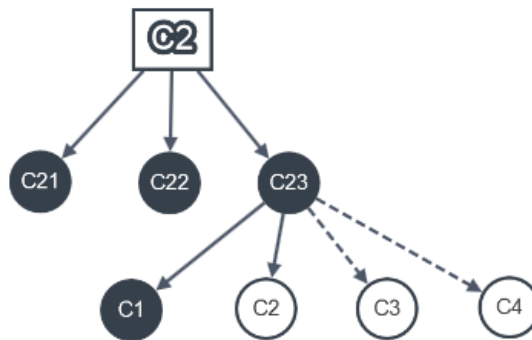


Рисунок 1.8 - Понятийный граф опорного понятия  $C_2$

Выражением знания для опорного понятия  $C_3$  - Требования предметной области, будет выражение:

$$C_3 \leq *C_{31} * C_{32} (*C_1 * C_2 + C_3); \quad (1.12)$$

а понятийный граф будет выглядеть следующим образом, как представлено на рисунке 1.9:



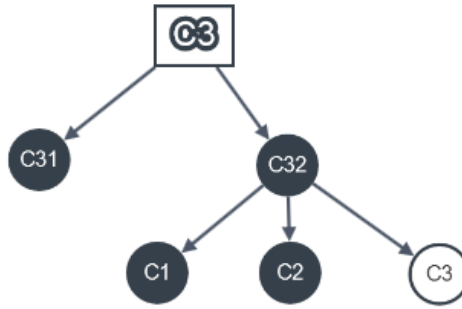


Рисунок 1.9 - Понятийный граф опорного понятия  $C_3$

Выражением знания для опорного понятия  $C_4$  - Системные требования, будет выражение:

$$C_4 \leq *C_{41} * C_{42} + C_{43}; \quad (1.13)$$

а понятийный граф будет выглядеть следующим образом, как представлено на рисунке 1.10:

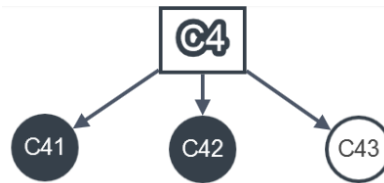


Рисунок 1.10 – Понятийный граф опорного понятия  $C_4$

Выражением знания для опорного понятия  $C_5$  – Нефункциональные требования, будет выражение:

$$C_5 \leq *C_{51} (*C_1 * C_2 * C_3 + C_4 * C_5 * C_6) * C_{52} (*C_1 + C_2 * C_3) * C_{53} (*C_1 + C_2 + C_3); \quad (1.14)$$

а понятийный граф будет выглядеть следующим образом, как представлено на рисунке 1.11:

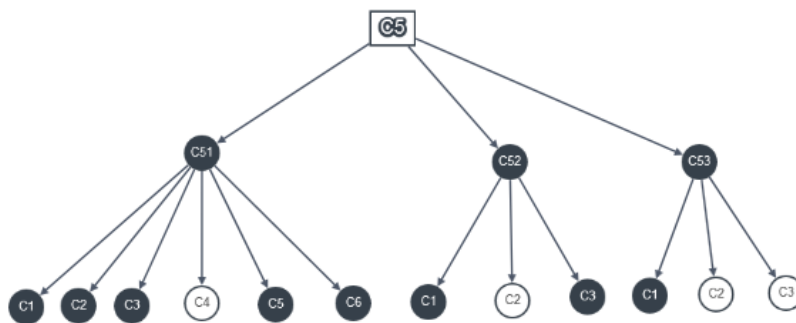


Рисунок 1.11 - Понятийный граф опорного понятия  $C_5$

Выражением знания для опорного понятия  $C_6$  - Аттестация требований, будет выражение:

$$C_6 \leftarrow *C_{61} + C_{62} * C_{63} * C_{64}; \quad (1.15)$$

а понятийный граф будет выглядеть следующим образом, как представлено на рисунке 1.12:

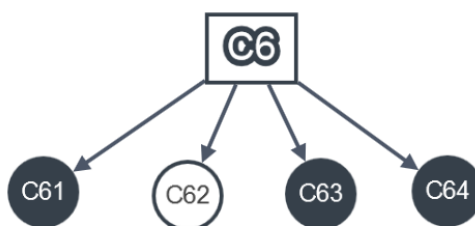


Рисунок 1.12 - Понятийный граф опорного понятия  $C_6$

Выражением знания для опорного понятия  $C_7$  - Управление требованиями, будет выражение:  $C_7 \leftarrow *C_{71} * C_{72}$ ;

а понятийный граф будет выглядеть следующим образом, как представлено на рисунке 1.13:

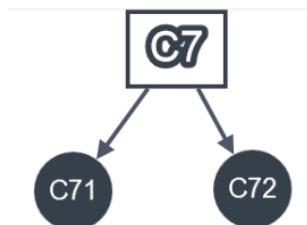


Рисунок 1.13 - Понятийный граф опорного понятия  $C_7$

Таким образом, выражение знания и выражение спецификации предназначены для представления онтологии опорного понятия и его характеристической модели в виде алгебраической модели, которую можно хранить в концептуальной базе данных и использовать при синтаксическом анализе и визуализации ориентированным графом.

Характеристическая модель позволяет изобразить количественные и качественные свойства понятия, что дает возможность разворачивать различные конфигурации опорного понятия. Введенные с этой целью отношения "Композиция", "Агрегация" и "Альтернативный выбор" позволяют отображать свойства общности и изменчивости на понятиях и на их характеристиках, и выделять повторно используемые образовательные компоненты предметной области, с целью конструирования новых знаний.

Отдельно следует отметить возможность отношений между опорными понятиями различных дисциплин, что позволяет демонстрировать механизмы взаимодействия таких понятий и конфигурировать новые знания и примеры реализации типовых решений при обучении.

В заключение данного раздела приведем очень интересный фрагмент из фундаментального труда "Порождающее программирование: методы, инструменты, применение", авторов Кшиштоф Чарнецки и Ульрих Айзенекер [48, с.96]: начало фрагмента - "Многие из тех, кто увлекается объектно-ориентированным программированием, не видят различий между понятиями и объектно-ориентированными классами. Они считают, что "все это относится к объектам". Это, безусловно, дилетантская идеология.

Между объектно-ориентированными классами и понятиями есть очевидное сходство. Объектно-ориентированные классы содержат общее описание ряда объектов. Аналогичным образом, понятия представляют собой общее описание ряда экземпляров понятий. Так в чем же разница между понятиями и объектно-ориентированными классами? Для того, чтобы найти ответ на этот вопрос, мы должны обратиться к уровню экземпляров. Объекты, то есть экземпляры объектно-ориентированных классов, обладают некоторыми предопределенными семантическими свойствами. У них есть состояние, они демонстрируют четко заданное поведение и обладают уникальной идентичностью. Напротив, у экземпляров понятий нет предопределенной семантики. Они могут быть чем угодно. Это различие проиллюстрировано на рисунке 1.14., где использовано отношение языка UML - наследование, причем направление стрелки вверх обозначает конкретизацию Понятия с помощью Объектно-ориентированного класса.



Рисунок 1.14 - Различие между понятием и объектно-ориентированным классом

Понятия можно представить как "ориентиры" нашего интеллекта, при помощи которых мы систематизируем явления. Каждое понятие олицетворяет некую категорию явлений. Важным понятиям следует присваивать имена, в противном случае нам придется перечислять все их свойства. Более того, понятия в значительной степени субъективны: их информационное наполнение зависит не только от субъекта, но также от времени, обстоятельств и других факторов." - конец фрагмента.

## 1.5 Визуализация онтологии и характеристической модели

Для визуализации онтологий и характеристических моделей адаптивной образовательной среды разработан парсер - специальный редактор, реализованный на объектно-ориентированном языке программирования С#, с использованием передового программного решения - Microsoft Visual Studio 2015.

Microsoft Visual Studio - это среда, позволяющая проектировать и создавать привлекательные приложения, удовлетворяющие самым взыскательным требованиям заказчиков.

По спецификации выражения знания для понятий, либо выражения спецификации для характеристической модели, собирается либо понятийный граф онтологии опорного понятия, либо характеристический граф. На рисунке 1.15 представлена диаграмма классов редактора в нотации языка объектного моделирования UML, а в Приложении А - листинг парсера.

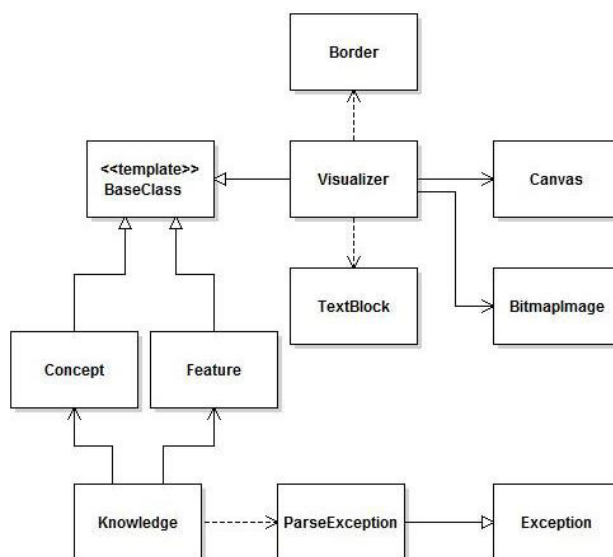


Рисунок 1.15 - Диаграмма классов редактора парсера адаптивной образовательной среды

С помощью интерфейса редактора можно выбирать модель представления, вводить выражение, проверять синтаксис спецификации выражения и, в случае его корректности, собирать модель с обязательной визуализацией модели в виде ориентированного графа.

Вид интерфейса редактора парсера, при успешной сборке онтологии по выражению знания, представлен на рисунке 1.16.

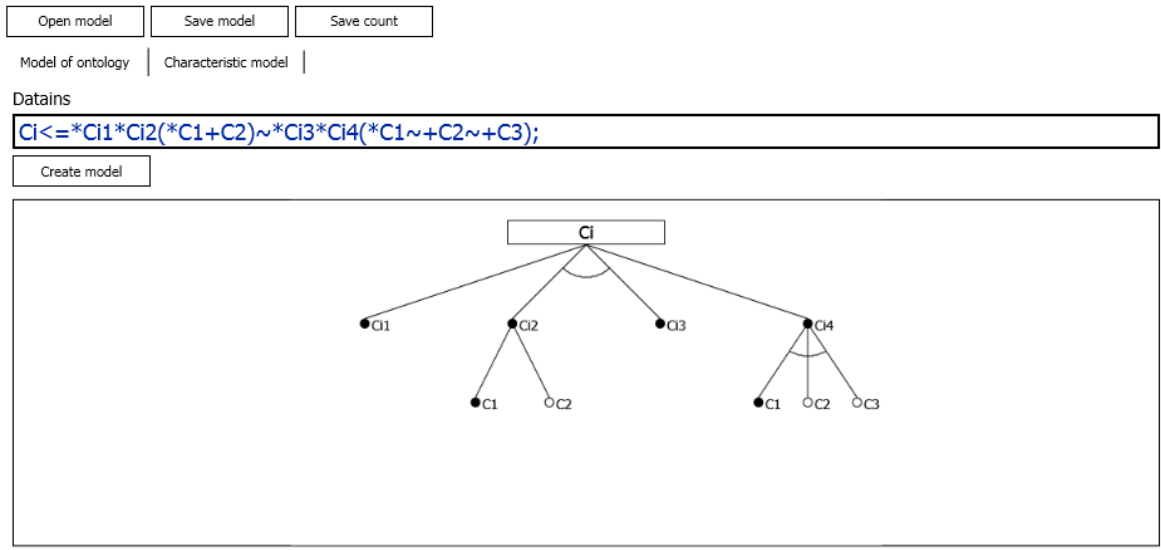


Рисунок 1.16 - Вид интерфейса парсера при успешной сборке онтологии

Возможные ошибки синтаксиса спецификации выражения могут быть сведены к следующим:

1. Нельзя конкретизировать необязательное понятие, так как оно является висячей вершиной.

2. Каждое конкретизирующее понятие либо характеристика должны быть помечены одним из основных символом: '\*' - композиция, '+' - агрегация операций языка спецификации, обозначающих соответственно обязательность и необязательность конкретизирующего понятия в экземпляре опорного понятия.

3. Присутствие символа '~' - тильда обязательно, когда рассматривается свойство изменчивости опорного понятия путем альтернативного выбора конкретизирующих понятий.

4. Поскольку в круглых скобках находится набор понятий второго уровня конкретизации, для которых контекстом является конкретизирующее понятие первого уровня, поэтому их индексация имеет простой цифровой символ.

5. Идентификация любого опорного понятия ограничена двумя уровнями конкретизации.

6. Каждое выражение знания, как и выражение спецификации, должно заканчиваться символом ';' - точка с запятой.

Например, в следующем выражении знания

$$C_i \leq *C_{i1} * C_{i2} (*C_1 + C_2) \sim *C_{i3} + C_{i4} (*C_1 \sim + C_2 \sim + C_3); \quad (1.16)$$

во вторых круглых скобках перечисляются конкретизирующие дочерние понятия для конкретизирующего понятия первого уровня  $C_{i4}$ , которое должно быть обязательным, а в выражении оно имеет статус - необязательная, что является недопустимой ситуацией (рисунок 1.17).



Рисунок 1.17 - Пример интерфейса редактора парсера, при обнаружении синтаксической ошибки в выражении знания

Построение онтологии понятий учебного материала можно представить как этапы выделения основных понятий, составляющих терминологический словарь предметной области; построения таксономии понятий; определения взаимосвязей между понятиями с помощью отношений "Композиция", "Агрегация", "Альтернативный выбор", идентификацию понятия с помощью выражения знания либо с помощью характеристической модели.

### Выводы по 1 разделу

По мере развития базы знаний содержания учебных дисциплин планируется расширение применения методики онтологического анализа учебного контента и организации процессов представления и организации знаний, на базе повторно используемых образовательных компонент.

Суть такой организации процесса онтологического инжиниринга заключается в том, чтобы он позволял представлять семантическое содержание образовательных средств и учебных ресурсов в виде коллекции абстрактных образовательных (знаниевых) компонент общего характера, из многочисленных сочетаний которых могут производиться эффективные образовательные траектории. Образовательные (знаниевые) компоненты должны быть ортогональными, не избыточными и предусматривать возможность повторного использования.

Применение принятой в наших исследованиях и построениях указанной методики организации процесса онтологического инжиниринга позволит создавать интегрированные учебные курсы совместных программ обучения в рамках реализации академической мобильности и возможности гибкого планирования учебного процесса.

В данном разделе изложены концепции инженерии знаний, теоретические основы формализации знаний, применение онтологического инжиниринга в представлении и организации семантических знаний предметной области; применение парадигм объектного и порождающего программирования, а также основные принципы инженерии предметной области, связанные с созданием абстрактных представлений. Показаны языковые конструкции, созданного для этих целей языка спецификации знаний, и их применение в рамках анализа общности и изменчивости понятий предметной области.

## **2 КОНЦЕПЦИЯ CDIO И ПРОЕКТНО - ОРИЕНТИРОВАННАЯ ОБРАЗОВАТЕЛЬНАЯ СРЕДА**

Всемирная инициатива CDIO - это проектно-ориентированная технология обучения, которая ориентирована на студента и интегрирована с проблемами и опытом реального производства.

Проектный подход и этапы всемирной инициативы CDIO следует рассматривать как контекст образовательной среды, который будет способствовать приобретению знаний, умений и практических навыков профессиональной инженерной деятельности обучающегося. В рамках такой образовательной среды должны происходить формирование комплексного подхода по определению образовательных потребностей студента и разработки последовательности учебных мероприятий, направленных на их удовлетворение.

Таким образом, ориентированная на проектный подход образовательная среда, определенная в рамках этапов *Conceiving, Designing, Implementing* и *Operating* инициативы CDIO, направлена на усвоение и глубокое понимание теоретических основ инжиниринга, применение которых будет способствовать приобретению знаний, практических навыков и умений, или, другими словами, на приобретение профессиональных компетентностей в будущей деятельности выпускника.

Для решения данной задачи большое значение должно отводиться применению современных педагогических подходов и инновационных методик представления и организации знаний, с целью:

- повышения когнитивной способности студентов;
- успешного усвоения необходимых знаний для решения практических учебно-производственных задач через призму этапов инициативы CDIO;
- усвоения обучающимися проектного опыта и активного применения полученных знаний в учебной и профессиональной деятельности.

Чтобы гарантировать достижение поставленных целей необходимо рассматривать инициативу CDIO как технологический процесс, в котором технология определения образовательных потребностей учитывала бы требуемый уровень компетентностей при выполнении практических учебно-производственных задач, и имела возможность моделирования и конструирования образовательных (знаниевых) компонент планируемого обучения для последующего проектирования дисциплин учебного плана.

Сформулированные таким образом требуемые результаты обучения служат основой для определения целей обучения, моделирования и последующего проектирования образовательных программ, процесса обучения и оценивания достижений студентов.

### **2.1 Компетентностный подход**

Таким образом, онтология компетентностного подхода отражает действие следующих принципов [23,24]:



Принцип 1. В качестве профессиональной деятельности человека проявляется качество всего человека, а не только его «профессиональный срез». Качество человека и качество профессионала образуют неразделимое единство.

Принцип 2. Принцип единства знаниевого и системодетельностного подходов. «Знания» и «деятельность» не являются оппозиционными категориями. Деятельность есть деятельностная актуализация знаний. Поэтому системодетельностный подход не противостоит знаниевому, а его дополняет.

Принцип 3. Принцип процессно-результатного дуализма. Этот принцип вытекает из «принципа отражения» в теории качества. Он определяет единство «процессно-ориентированного» и «результатно - ориентированного подхода». Компетенция как результат формируется системой процессов образования, обучения и воспитания, отражающих определенные модули содержания образования.

Принцип 4. Принцип изоморфизма между структурами системы видов деятельности, идентифицирующей модель качества выпускника вуза, системы квалификационных требований, системы классов задач профессиональной деятельности и системой компетенций в процессе системодетельностного и компетентностного моделирования качества подготовки выпускников вуза.

Принцип 5. Принцип изоморфизма между системодетельностно - компетентностной моделью качества подготовки выпускника вуза и предметно-дисциплинарной (модульной) моделью качества подготовки выпускника вуза.

Таким образом, компетенция выступает компонентом потенциального качества подготовки выпускника вуза, выражающим собой его способность к выполнению определенного комплекса задач или вида деятельности.

Компетентностный подход как определенный вид формализации имеет свои ограничения и недостатки и не может рассматриваться как доминирующий для описания качества подготовки выпускника вуза, а только как дополнительный.

Его ограниченность состоит в том, что он не может адекватно отразить в себе богатство внутреннего мира человека, не охватывает собой процесс воспитания и социализации личности как целостность. Ограниченность компетентностного подхода состоит также и в том, что он обращен к отдельному выпускнику вуза, в то время, как он должен включать в себя системно-интегративный результат в виде роста качества общественного интеллекта (качества интеллектуальных ресурсов общества) и роста качества человека, и на их основе – роста качества жизненных сил в обществе.

Компетентностный подход не может мотивироваться в своих целевых установках принципом рыночного фундаментализма и исходить из интересов глобализации рынка образования и труда, поскольку эта тенденция является с позиций онтологии человечества утопичной, ведет к отрыву выпускников вузов от «корневых», ценностных оснований родной культуры, формирует из них «неокочевников», и фактически ускоряет процесс отчуждения человека от природы, его дальнейшее движение к экологической гибели.

Компетентностный подход должен обеспечивать опережающее развитие качества человека, качества общественного интеллекта и качества образовательных систем в обществе, фундаментальную и универсальную подготовку выпускников вузов с широким интегративным мировоззрением и духовно-нравственными началами.

Какая же категория является ведущей в компетентностном подходе с точки зрения его реализации – компетенция или компетентность?

Компетенция есть «системное образование» в личности обучающегося, которое является «компонентом его качества». Причем компетенция относится в большей степени к потенциальному качеству, чем к актуальному качеству, поскольку актуализация в образовательном процессе в форме лабораторных и практических занятий, профессиональных практик, и пр. несет не полный характер и не может заменить собой полностью будущую профессиональную деятельность выпускника. Поэтому выпускник в процессе обучения получает комплекс компетенций, который и только условно можно считать компетентностью, или компетентностью начального уровня.

Компетентность есть актуализированный комплекс компетенций, она относится к категории актуального или реального качества и формируется главным образом за пределами учебного заведения по мере накопления опыта в соответствующей профессиональной деятельности.

Компетентность есть динамическое качество человека- профессионала, которое «движется» от начального уровня, заложенного в системе высшего профессионального образования, к мастерству, как высшей форме компетентности.

Поэтому компетенция является главной категорией компетентностного подхода для экспликации качества профессионального образования, а компетентность – вторичной, производной категорией.

#### Знания

Знания определяют содержание компетенции и компетентности. Как «умения» и «навыки», так и «компетенции» и «компетентность» являются формами инструментализации знаний. «Владеть» чем-то с точки зрения системодейственного подхода – означает «знать», которое превратилось в «уметь» и иметь соответствующий «навык».

Компетентностная форма представления профессионального качества имеет иерархическую структурность и может быть описана «деревом» или «графом» компетенций.

Иерархичность системы компетенций определяет вложенность компетенций, означающий, что одни компетенции могут входить в другие, более высокого уровня, и становиться их компонентами.

Нижний уровень «дерева компетенций» отражает единичные компетенции, как условно неделимые. Узлы дерева компетенций могут трактоваться как групповые или сложные компетенции. Если единичные компетенции репрезентируются каким-то классом задач или функцией деятельности, то

групповые компетенции репрезентируются комплексом задач, сложной функцией или видом деятельности.

Особое значение в эпистемологии компетентностного подхода принадлежит категории «знание». Знание – сложная категория, имеющая смысловую многозначность и следующие определения:

- знание есть модель (образ) объективной реальности, позволяющая осуществлять ее экспликацию и прогнозирование развития «реальности»;
- знание есть часть информации, которая прошла процедуру узнавания (распознавания), классификации, понимания (осмысления);
- знание есть результат познания. В соответствии с принципом объект - процессного дуализма в знании реализуется одновременно «знание процесс» и «знание – результат»;
- знание есть результат учения, образования. Трансляция знаний – процесс, пронизывающей всю триаду образовательного процесса: воспитание, обучение, образование как передачу знаний в узком смысле слова;
- знание в отличие от всех способов отражения мира, предполагает сознание, т.е. осознание субъектом мира таким, как он сам существует сам по себе, и образ этого мира.

Компетенция является формой бытия знания, в том плане, что компетенция является моделью поведения человека при решении определенного класса задач, осуществления ролей или функций в определенных ситуациях профессиональной деятельности.

В качестве важнейших компетенций были выделены:

- способность к анализу и синтезу;
- способность учиться;
- способность решать проблемы;
- способность применять знания на практике;
- способность приспосабливаться к новым ситуациям;
- забота о качестве;
- способность управлять информацией;
- способность работать самостоятельно в команде.

Из этого следует, что фактически в проекте TUNING подчеркнут приоритет фундаментальной подготовки выпускников вузов, в том числе системной подготовки и способности к самообучению. Таким образом, компетентностный подход имеет свою нишу в системной эпистемологии образовательных систем.

Эпистемологическая функция компетентностного подхода многообразна, она включает в себя:

- усиление ориентированности образования на трудоустройство, т.е. на повышение конкурентоспособности выпускников вуза на рынках труда;
- на увеличение гибкости высшего образования на основе обеспечения изоморфизмов знаниевого (предметно-дисциплинарного) и компетентностного подходов на учебно-модульной основе;

– повышение измеримости качества выпускников вуза и обеспечение совместимости результатов подготовки выпускников по одному профилю профессиональной деятельности в разных вузах и разных странах.

Появление категории компетенции сопровождалось одновременным процессом их классификации (систематизации, типологизации).

Классификация компетенции есть раскрытие содержания категории компетенции на основе движения от общего к конкретному и единичному.

В рамках «Проекта TUNING» компетенции формируемые высшим профессиональным образованием, разбиты на две категории: компетенции, относящиеся к предметной области – предметно-специализированные компетенции и универсальные компетенции (навыки), не связанные непосредственно с решением профессиональных задач, но связанные с успешностью профессиональной деятельности опосредованно (рисунок 2.1):



Рисунок 2.1- Дерево классов компетенций в версии проекта TUNING

Предметно-специализированные компетенции по проекту TUNING отражают ключевые компетенции», которые тесно связаны со специфическим знанием области обучения и обеспечивают своеобразие и состоятельность конкретных программ на соискание академической степени. Другими словами, предметно-специализированные компетенции отражают на уровне навыков, умений, способностей выпускников специфику профессиональной части образовательных программ.

Универсальные компетенции – это те компетенции, которые по проекту TUNING считаются важными для конкретных социальных групп выпускников и работодателей. Иными словами, различие между универсальными и предметно-специализированными компетенциями в мощности соответствующей категории компетенций: мощность универсальных

компетенций такова, что они являются общими и, в такой же мере обязательными, почти для всей высшей школы Европы, а предметно-специализированные компетенции отражают их адресность только к определенным конкретным программам на соискание степени бакалавра или магистра.

В целом деление компетенций на универсальные и предметно-специализированные компетенции отражает принятое деление на общеобразовательную и профессиональную подготовку, которые могут быть, в компетентностном формате, представлены в виде и базовых и профессиональных компетенций.

Таким образом, можно констатировать, что на Западе действует множество целевых классификаций компетенций/компетентностей, выстраиваемых исходя из разных целей управления (политики) и из разных ситуаций оценивания (диагностики).

В России за последнее время предложено значительное количество классификаций компетенций. Например, в проекте нового поколения государственного образовательного стандарта предложено трактовать:

– компетентность как готовность выпускника определенной ступени образования к выполнению видов и задач профессиональной деятельности с использованием полученных знаний, умений и навыков, при ясном понимании их социальной значимости, а также связанных с ними социальных последствий;

– компетенция как готовность студента к применению на практике междисциплинарных знаний и умений, полученных при изучении циклов дисциплин и при ясном понимании их значимости для профессиональной подготовки [28].

В соответствии с принятой классификацией в ГОС [7] предложены следующие деревья классов компетенций бакалавра (рисунок 2.2) и магистра (рисунок 2.3):

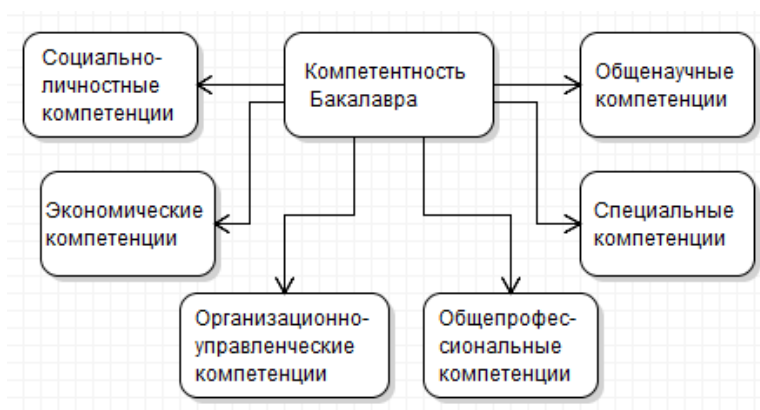


Рисунок 2.2 - Дерево компетенций бакалавра

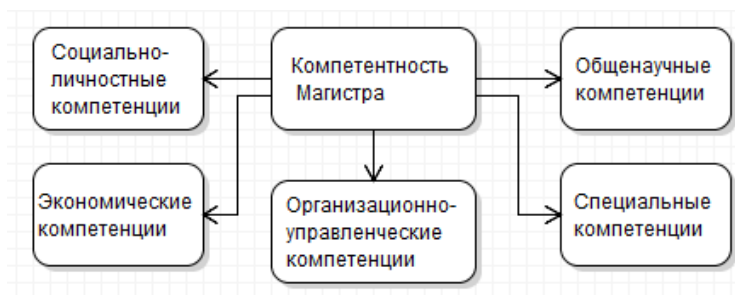


Рисунок 2.3 - Дерево компетенций магистра

Каждый из этих классов компетенций раскрывается на уровне категорий «должен», «понимать», «знать», «владеть», «уметь», «иметь научное представление», «быть готовым», «обладать способностью», «стремиться», «иметь целостное представление», «использовать», «быть подготовленным», «иметь опыт».

Таким образом, изложенный анализ показывает широкий спектр подходов к классификации компетенций, что можно зафиксировать в виде следующих методологических положений [49,50]:

Положение 1. Существует множество целевых классификаций компетенций, объективно обусловленных разными целевыми (ценностными) установками и используемых оснований.

Положение 2. Множество различий в классификациях компетенций обусловлено использованием разных понятийных (семантических) систем для их описания.

В значительной части это обусловлено тем, что сама проблема такой классификации носит философско-методологический, научно-междисциплинарный характер и занимаются этой проблемой узкие специалисты - эксперты, исходящие в своей семантике из здравого смысла.

Положение 3. Множество классификаций компетенций будет сужаться по мере роста их адекватности в отражении качества человека и качества профессионала (специалиста).

Таким образом, главным направлением в развитии высшей школы человечества в XXI веке является фундаментализация высшего образования, ее ноосферизация, реализация императивов «человеческой революции». Образование в целом, и высшая школа в первую очередь, должны стать главным механизмом восходящего воспроизводства качества человека, качества общественного интеллекта и качества самих образовательных систем в обществе. Это главное условие устойчивого развития в XXI веке и обеспечения выхода человечества из первой фазы Глобальной экологической Катастрофы.

## **2.2 Проектно-ориентированное инженерное обучение по направлению "Инженерия программного обеспечения"**

Проектно-ориентированная образовательная среда - это образовательный контекст, способствующий приобретению базовых технических знаний

программной инженерии, используемых для планирования, проектирования, производства и применения процессов и технологий при разработке программных продуктов и систем.

В проектно-ориентированной образовательной среде обучения, по направлению "Инженерия программного обеспечения", выделяются:

- области знаний SWEBOOK, определяющие базовые инженерные знания программной инженерии;
- CDIO стандарты, в которых конкретные результаты обучения представлены в виде подробного перечня компетенций или квалификационных требований инженера;
- теоретические основы инжиниринга, связанные с процессами создания и эксплуатации инженерных объектов и систем;
- комплексная технология определения образовательных ресурсов и учебных мероприятий, как основа планируемого обучения CDIO Syllabus.

Поскольку система академического образования в высшем учебном заведении должна быть сбалансированной, устойчивой и динамично развивающейся системой подготовки высококвалифицированных кадров, поэтому принципиально важным является создание такой адаптивной образовательной среды, которая позволяла бы генерировать и воспроизводить семантические знания, адекватные актуальным научно-технологическим вызовам современного развития общества и экономики.

К семантическим знаниям мы относим знания об определяющих свойствах связанных с компетентностными моделями, как абстрактных конструкциях высокого уровня, которые имеют свою организацию, структуру и автономное представление.

В качестве модели семантических знаний каждого из этапов инициативы CDIO будем считать совокупность необходимых и достаточных опорных понятий, определенных в рамках профессиональных, базовых и дополнительных компетенций. Опорные понятия семантических знаний, в свою очередь, предполагают свою идентификацию с помощью конкретизирующих понятий первого и второго уровней и последующей спецификацией выражениями знания.

Таким образом, представление о специальности "Вычислительная техника и программное обеспечение" направления "Инженерия программного обеспечения" можно определить:

- профессиональными профилями специальности;
- этапами инициативы CDIO;
- ядром знаний SWEBOOK;
- компетентностными моделями, представляемыми агрегацией профессиональных, базовых и дополнительных компетенций соответствующих этапов инициативы CDIO;
- онтологиями опорных понятий компетентностных моделей, в которых понятия наделены свойствами общности и изменчивости;

– характеристическими моделями, с помощью которых специфицируются количественные свойства понятий.

Под образовательной траекторией обучения (EL - Education Line), будем понимать некоторую структуру из множества компонентов обучения (LC - Learning Component), обеспечивающих технологические процессы разработки и производства инженерных объектов и систем в рамках каждого из этапов инициативы CDIO.

$$EL = (LC_1, LC_2, \dots, LC_m), \quad (2.1)$$

где компонентом обучения  $LC = (O_m, F_m)$  является компонент, состоящий из моделей представления семантических знаний. При формировании компонентов обучения специальности учитываются концепции общности и изменчивости свойств понятий, что позволяет осуществлять повторное использование таких компонентов как при проектировании учебных дисциплин специальности, так и при моделировании образовательных траекторий.

Предметными областями специальности "Вычислительная техника и программное обеспечение", направления "Инженерия программного обеспечения", являются практически все сферы человеческой деятельности, связанные с процессами и технологиями разработки программных продуктов и систем. Под доменом предметной области будем понимать пространство, в котором формулируется определенная задача, характеризующаяся соответствующим набором понятий, каждое из которых представляет важный аспект для решения учебно-производственной задачи в рамках этапов инициативы CDIO. Домен предметной области используется для моделирования и описания семантических знаний предметной области и формирования онтологий.

Инициатива CDIO имеет три общих целей обучения, позволяющие студентам продемонстрировать глубокие технические знания и мышление по специальности; практическое мастерство в создании и эксплуатации новых продуктов и систем; понимание важности и стратегического значения научно-технического развития информационных технологий и общества.

### **2.3 Компетентностные модели и этапы CDIO**

Совершенствование качества подготовки специалистов должно основываться на компетентностном подходе, рассматриваемом с точки зрения профессиональных компетенций, из которых выделим базовые, как основа профессии и дополнительные, принадлежащие другим типам деятельности, но значимые для определенного рабочего места или профиля выпускника.

Для каждого из этапов инициативы CDIO определим уровни профессиональных компетенций, в рамках каждого из которого определяются базовые и дополнительные компетенции. При этом:

– профессиональные компетенции, для любого этапа инициативы CDIO, являются обязательными по определению;



– базовые и дополнительные компетенции, для любого этапа инициативы CDIO, могут быть обязательными, необязательными и альтернативными компетенциями;

– профессиональные, базовые и дополнительные компетенции описываются с помощью онтологий опорных понятий и соответствующими выражениями знания, где опорные понятия представляют собой семантические знания предметной области;

– идентификация опорных понятий профессиональных, базовых и дополнительных компетенций происходит с помощью конкретизирующих понятий.

Продемонстрируем спецификацию профессиональных, базовых и дополнительных компетенций с помощью разработанного для этих целей языка представления знаний.

Пусть для некоторого профессионального профиля специальности "Вычислительная техника и программное обеспечение", направления "Инженерия программного обеспечения" необходимо освоение профессиональных (ПК), базовых (ВК), и дополнительных компетенций (ДК).

Допустим, что для успешной реализации ключевых вопросов, возникающих в процессе разработки проекта, для каждого из этапов инициативы CDIO, требуются знания профессиональных, базовых и дополнительных компетенций, представленные следующими выражениями компетенций:

- $\text{Conceiving} \leq *PK_1(*P_1*P_2)*PK_2*PK_3;$
- $\text{Conceiving} \leq *BK_1 \sim +BK_2*BK_3(*B_1 \sim +B_2)+BK_4;$
- $\text{Conceiving} \leq *DK_1+DK_2*DK_3 \sim +DK_4;$
- $\text{Designing} \leq *PK_1*PK_2*PK_3;$
- $\text{Designing} \leq *BK_1+BK_2*BK_3;$
- $\text{Designing} \leq *DK_1 \sim *DK_2(*D_1+D_2) \sim *DK_3;$
- $\text{Implementing} \leq *PK_1*PK_2*PK_3*PK_4(*P_1*P_2)*PK_5;$
- $\text{Implementing} \leq *BK_1*BK_2*BK_3(*B_1 \sim +B_2) \sim +BK_4;$
- $\text{Implementing} \leq *DK_1*DK_2(*D_1+D_2)*DK_3+DK_4;$
- $\text{Operating} \leq *PK_1*PK_2;$
- $\text{Operating} \leq *BK_1(*B_1*B_2)*BK_2+BK_3;$
- $\text{Operating} \leq *DK_1(+D_1 \sim +D_2)*DK_2+DK_3;$

где в круглых скобках специфицированы соответствующие компетенции второго уровня конкретизации. Заметим, что профессиональные компетенции - ПК, являются обязательными компетенциями этапа, а базовые - ВК и дополнительные - ДК, могут быть обязательными, необязательными и альтернативными компетенциями этапа инициативы CDIO.

Каждая из компетенций, формализованная в виде набора опорных понятий и их выражений знаний, определяет семантические знания, освоение которых необходимо и достаточно как при теоретической подготовке, так и для решения

практических учебно-производственных задач каждого из этапов инициативы CDIO.

Например, профессиональную компетентность  $PK_1$  этапа *Conceiving* можно представить следующим набором обязательных опорных понятий:  $PK_1 \leq *C_1 * C_2 (*C_1 * C_2) * C_3 * C_4$ ;

где в круглых скобках вводятся обязательные понятия второго уровня конкретизации, для которых семантика определяется контекстом соответствующего им опорного понятия.

Далее, каждое из опорных понятий компетенции можно представить выражением знания, включающим обязательные, необязательные, либо альтернативные понятия первого и второго уровней конкретизации. Например, опорное понятие  $C_1$  профессиональной компетенции  $PK_1$  можно идентифицировать следующим набором понятий первого и второго уровней конкретизации:  $C_1 \leq *C_{1.1} * C_{1.2} (*C_1 \sim + C_2) \sim *C_{1.3} + C_{1.4}$ ;

С помощью таких выражений знаний формируются образовательно-знаниевые компоненты, а поскольку в выражениях знаний предусмотрено использование свойств общности и изменчивости понятий, то такие компоненты можно повторно использовать в качестве учебного контента для проектирования учебных дисциплин.

Пусть успешное усвоение знаний этапа *Conceiving*, дает возможность обучающемуся овладеть профессиональными компетенциями:  $PK_1$ ,  $PK_2$ ,  $PK_3$  и  $PK_4$ . Представим такую ситуацию следующим образом:  $Conceiving \leq *PK_1 * PK_2 * PK_3 * PK_4$ ; и, визуально, следующим ориентированным графом (рисунок 2.4):

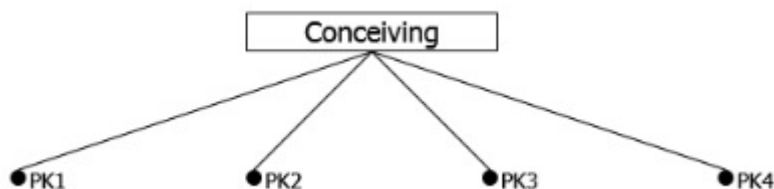


Рисунок 2.4 - Представление этапа *Conceiving* профессиональными компетенциями

Приведенное представление показывает, что на этапе *Conceiving* у студента должно быть сформировано четыре обязательных профессиональных компетенции:  $PK_1$ ,  $PK_2$ ,  $PK_3$  и  $PK_4$ .

Дальнейшее уточнение профессиональных компетенций допускает их описание с помощью базовых и дополнительных компетенций.

Например, успешное усвоение знаний этапа *Conceiving*, обеспечивает обучающегося двумя обязательными базовыми компетенциями:  $BK_1$ ,  $BK_2$ , и одной необязательной -  $BK_3$ , что можно представить следующим выражением компетентностей:  $Conceiving \leq *BK_1 * BK_2 + BK_3$ ;

Если, при этом, профессиональные компетенции сопровождаются еще и дополнительными, тогда наше представление будет иметь следующий вид:  
 $Conceiving \leq *DK_1(*D_1+D_2)*DK_2\sim+DK_3;$

Таким образом, этап *Conceiving*, представлен через призму необходимых и достаточных профессиональных, базовых и дополнительных компетенций.

Далее, для каждой из выделенных компетенций, следует их определение с помощью опорных и конкретизирующих понятий.

Например, для приобретения базовой компетенции  $BK_1$  следует изучить опорные понятия, спецификация которых представлена выражением:  
 $BK_1 \leq *C_1(*C_1\sim*C_2*C_3)*C_2*C_3(*C_1*C_2);$

и визуально продемонстрированное на рисунке 2.5.

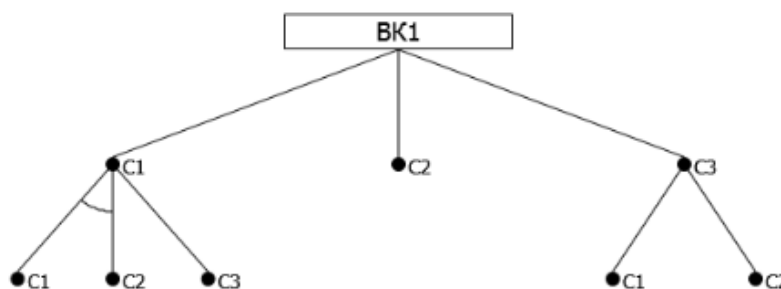


Рисунок 2.5 - Понятийный граф базовой компетенции  $BK_1$

И, наконец, каждое опорное понятие, например компетенции  $BK_1$ , идентифицируем конкретизирующими понятиями первого и второго уровней. Например, опорное понятие  $C_1$ , являющееся необходимым и достаточным для приобретения базовой компетенции  $BK_1$  этапа *Conceiving*, можно представить следующим выражением знания:  $C_1 \leq *C_{1.1}*C_{1.2}(+C_1*C_2)\sim+C_{1.3}*C_{1.4};$

а соответствующий понятийный граф будет иметь вид, представленный на рисунке 2.6:

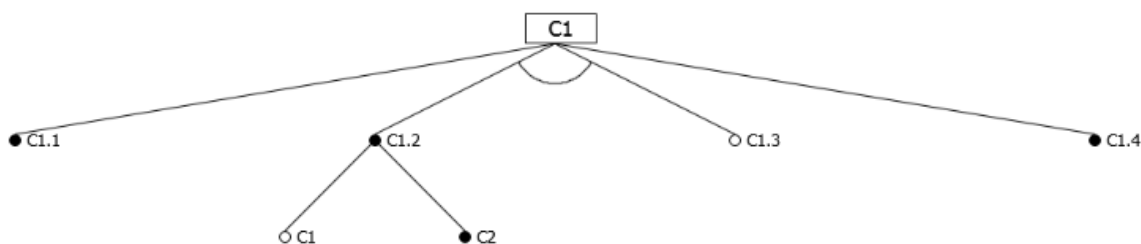


Рисунок 2.6 - Понятийный граф опорного понятия  $C_1$  базовой компетенции  $BK_1$

## 2.4 Моделирование образовательных траекторий обучения

В соответствии с национальной рамкой квалификаций и профессиональными стандартами Дублинских дескрипторов и Европейской рамкой квалификаций, вуз самостоятельно разрабатывает различные образовательные программы.

Дублинские дескрипторы представляют собой описание уровня и объема знаний, умений, навыков и компетенций, которые должны приобрести студенты по завершении образовательной программы каждого уровня (ступени) высшего и послевузовского образования, базирующиеся на результатах обучения, сформированных компетенциях и общем количестве кредитных единиц ECTS.

Поэтому рекомендуется в рамках каталога элективных дисциплин специальности представлять на выбор студентам несколько образовательных траекторий, то есть перечней элективных дисциплин и последовательности их изучения, позволяющих студенту «на выходе» освоить образовательную программу, ориентированную на конкретную сферу деятельности с учетом потребностей рынка труда.

Для моделирования образовательной траектории обучающегося, в соответствии с требуемыми компетенциями, необходимо располагать образовательно-знаниевыми компонентами в формате онтологий и выражений знаний.

В качестве исходных базовых концепций для формирования образовательно-знаниевых компонент приняты:

- ядро знаний SWEBOOK - нормативный документ IEEE в области программной инженерии,
- концепция проектного подхода,
- профессиональная компетентностная модель выпускника,
- концепции всемирной инициативы CDIO,
- концепция общности и изменчивости,
- онтологический инжиниринг,
- представление и организация знаний на основе онтологии.

Все перечисленные базовые концепции образовательной среды, с учетом синтаксиса отношений языка объектного моделирования UML, представлены на рисунке 2.7.

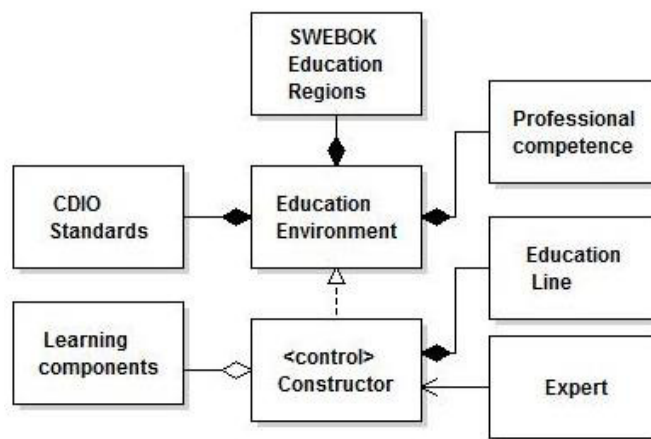


Рисунок 2.7 - Компоненты образовательной среды

Центральным компонентом является образовательная среда, агрегирующая концепции: стандартов CDIO, области знания инженерии программного обеспечения - SWEBOK и профессиональные компетенции, а реализующим компонентом является конструктор эксперта, с помощью которого агрегируются образовательные (знаниевые) компоненты среды, в соответствии с требуемой образовательной траекторией обучения.

В настоящее время для инженерных специальностей инициируется проектный подход, согласно которого обучение должно строиться так, чтобы обеспечить студента всем тем, что необходимо ему в его профессиональной деятельности, в соответствии со всеми аспектами процессов создания продуктов и систем. В тоже время такое обеспечение должно определять планирование и организацию учебного процесса и учебных мероприятий, а также условия и ограничения, связанные с подготовкой высокопрофессиональных специалистов.

CDIO - это проектно-ориентированная технология обучения, интегрированная с проблемами и опытом реального производства.

Результаты обучения определяются на основе Дублинских дескрипторов первого уровня (бакалавриат) и формулируются как на уровне всей программы, так и на уровне модуля и отдельной дисциплины.

Дескрипторы первого уровня предполагают способности выпускника:

1) демонстрировать знания и понимание, включая элементы наиболее передовых знаний предметной области, и применять эти знания и понимание на профессиональном уровне;

2) аргументировано формулировать и решать проблемы в изучаемой предметной области;

3) осуществлять сбор и интерпретацию информации в изучаемой предметной области для формирования суждений с учетом социальных, этических и научных соображений;

4) сообщать информацию, идеи, проблемы и решения, как специалистам, так и не специалистам.

Для формирования индивидуального учебного плана студента, используется типовой учебный план и каталог элективных дисциплин. Каталог элективных дисциплин представляет собой систематизированный аннотированный перечень всех дисциплин компонента по выбору и их краткое описание, с указанием цели изучения, краткого содержания и ожидаемых результатов, то есть знаний, умений, навыков и компетенций приобретаемых студентами.

Проектный подход, следуя этапам CDIO, рассматривается через призму профессиональных, базовых и дополнительных компетенций, представляемых онтологиями опорных понятий и выражениями знаний, и усвоение которых гарантирует приобретение навыков решения практических и учебно-производственных задач.

Ядром образовательно-знаниевого компонента среды является знаниевая модель из опорных понятий предметной области вместе с выражениями знаний. Образовательно-знаниевый компонент используется для формирования учебного контента и для проектирования дисциплин специальности.

В настоящее время практика планирования учебного плана специальности происходит детерминированным образом, то есть разработка рабочего учебного плана и соответствующих учебно-методических комплексов дисциплин осуществляется на базе типового учебного плана и элективных дисциплин, что ограничивает возможности гибкого планирования как дисциплин, так и оперативного изменения учебного контента дисциплин, связанных с постоянным развитием технологий и с изменениями требований работодателей и социальными запросами общества.

Поэтому в рамках разработки студенческих проектов профессионального профиля специальности, и соответствующих квалификаций, а также с учетом требований работодателей и социальным запросом общества, необходимо:

- определить компетентностные модели как композиции профессиональных, базовых и дополнительных компетентностей, необходимых и достаточных для решения практических и учебно-производственных задач каждого из этапов инициативы CDIO;

- в рамках каждой из компетентностных моделей определить наборы опорных понятий, представляющие собой семантические знания этапов инициативы CDIO, знания которых необходимы для выполнения учебно-производственных задач студенческого проекта:  $PK = C1, C2, \dots$ ,  $BK = C1, C2, \dots$ ,  $DK = C1, C2$ ,

- каждое из опорных понятий компетентностных моделей представить онтологиями и выражениями знаний.

Разработанный для этих целей язык описания знаний можно использовать для спецификации свойств общности и изменчивости опорных и конкретизирующих понятий, что позволяет конфигурировать образовательно-знаниевые компоненты для проектирования новых дисциплин специальности.

Допустим, необходимо сформировать индивидуальную образовательную траекторию обучения студента, или учебный план, либо надо произвести

корректировку и изменения в дисциплинах учебного плана специальности по некоторому набору профессиональных, базовых и дополнительных компетенций.

Для этих целей формируются онтологии опорных понятий, каждая из которых представляется своими экзеплярами, или другими словами, траекториями различных вариантов конкретизирующих понятий.

Таким образом, задача заключается в том, чтобы для выбранного профессионального профиля специальности и в соответствии с требуемыми профессиональными компетенциями этапов инициативы CDIO по разработке студенческого проекта, обосновать и сформировать образовательно-знаниевые компоненты с помощью которых представлять и организовывать знания, необходимые и достаточные для решения практических и учебно-производственных задач и приобретения обучающимися профессиональных навыков и умений, и которые станут основой учебного контента проектируемых дисциплин учебного плана специальности, с учетом условий и ограничений кредитной технологии обучения.

Такие представления знаний позволят проектировать и разрабатывать, в первую очередь, дисциплины профильного цикла, а что касается базовых дисциплин, то они должны будут формироваться путем реализации базовых и дополнительных компетенций и необходимости в них для профилирующих дисциплин специальности.

Модели онтологий студенческого проекта

Согласно концепции инициативы CDIO любой технологический процесс выполнения студенческого проекта представляется последовательностью четырех обязательных этапов:

Студенческий проект = { Задумка - *Conceiving*, Проектирование-*Designing*, Реализация-*Implementing*, Управление и применение-*Operating* }.

На первом шаге технологического процесса любого из этапов инициативы CDIO определяются компетенции или, другими словами, наборы знаний, которыми должны овладеть студенты для успешного выполнения учебных мероприятий и решения практических и учебно-производственных задач, связанных с разработкой студенческого проекта.

Пусть для успешного выполнения проектных работ этапа *Conceiving*, обучающимся необходимо усвоить знания компетентностной модели этапа, которые представлены онтологиями опорных понятий, конкретизирующими понятиями и соответствующими выражениями знаний.

1) Компетентностная модель каждого их этапов инициативы CDIO представляется выражением, из последовательности профессиональных, базовых и дополнительных компетенций.

Например, для этапа *Conceiving* компетентностную модель представим следующим выражением компетентностей:  $\text{Conceiving} \leq *PK_1 * PK_2 * PK_3 * BK_1 * BK_2 + BK_3 + BK_4 * DK_1 * DK_2$ ; и графически, как на рисунке 2.8

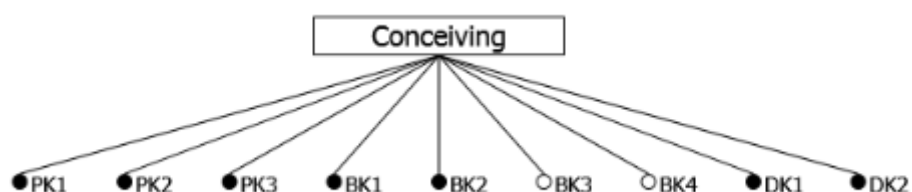


Рисунок 2.8 - Компетентностная модель этапа Conceiving

2) Каждый из элементов компетентностной модели описывается наборами опорных и конкретизирующих понятий и соответствующими выражениями знаний. Причем, профессиональные компетентности должны представляться только обязательными опорными понятиями, а базовые и дополнительные, помимо обязательных, могут представляться необязательными и альтернативными понятиями, другими словами, включать в свои описания концепцию изменчивости.

3) Далее, каждое опорное понятие идентифицируем конкретизирующими понятиями первого и второго уровней. Например, обязательное опорное понятия  $C_3$  профессиональной компетенции  $PK_2$  можно представить следующим выражением знания и графически, как на рисунке 2.9:  $C_3 \Leftarrow *C_{3.1} * C_{3.2} * C_{3.3} * C_{3.4} (+C_1 \sim +C_2)$ ;

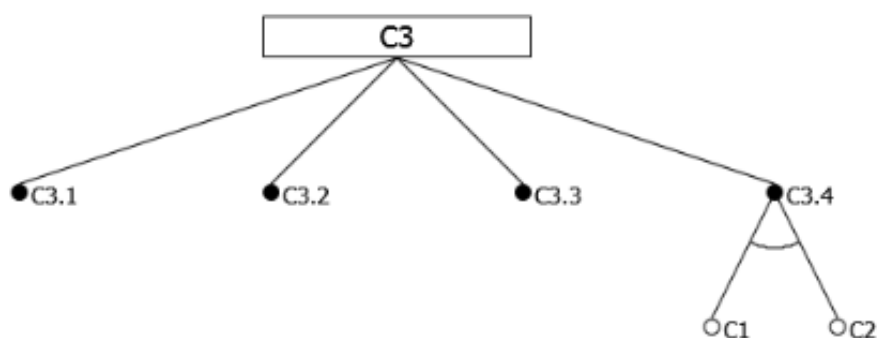


Рисунок 2.9 - Понятийный граф обязательного опорного понятия  $C_3$  профессиональной компетенции  $PK_2$

в котором допускается возможность изменчивости за счет присутствия необязательных и альтернативных конкретизирующих понятий.

4) Все, таким образом организованные представления знаний можно отобразить в виде иерархии понятий, каждая из траекторий которой является некоторым вариантом знания, то есть набором из конкретизирующих понятий, необходимых и достаточных как для выполнения учебных заданий и мероприятий, так и для реализации проектных решений по студенческому проекту.

Напомним, что CDIO - это проектно-ориентированная технология обучения студента, интегрированная с проблемами и опытом реального производства, в которой технологический процесс выполнения студенческого



проекта, согласно концепции инициативы CDIO, должен быть представлен последовательностью четырех обязательных этапов: Conceiving - Задумка; Designing - Проектирование; Implementing - Реализация; Operating - Управление и применение.

Рассмотрим модельные примеры возможных вариантов представления и организации знаний профессиональных, базовых и дополнительных компетенций.

1) Пусть профессиональная компетентность РК обусловлена изучением следующих обязательных опорных понятий:  $C_1, \dots, C_i, \dots, C_n$ .

2) Пусть для идентификации опорного понятия  $C_i$  профессиональной компетенции РК, требуются два обязательных понятия первого уровня -  $C_{i1}, C_{ij}$ , одно необязательное понятие -  $C_{in}$ , и три конкретизирующих понятий второго уровня -  $C_1, C_2, C_3$ , из которых понятие  $C_1$  - обязательное,  $C_2$  - необязательное, а  $C_3$  - альтернативное понятие. Таким образом сформировали следующее выражение знания:  $C_i \leq *C_{i1} * C_{ij} (*C_1 + C_2 \sim + C_3) + C_{in}$ ; понятийный граф которого будет иметь вид (рисунок 2.10):

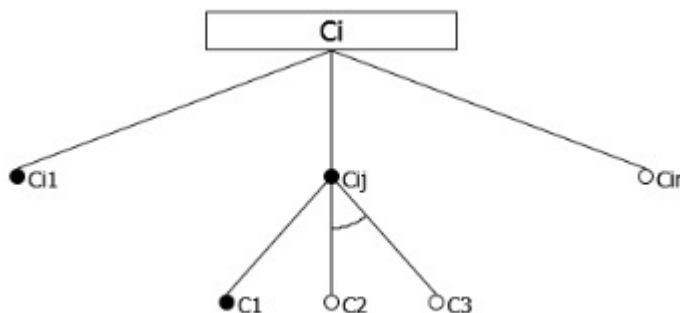


Рисунок 2.10 - Понятийный граф опорного понятия  $C_i$  профессиональной компетенции РК

3) Базовые и дополнительные компетенции обусловлены изучением не только обязательных опорных понятий, но могут допускать изменчивость в их представлениях уже на первом уровне идентификации, с помощью необязательных и альтернативных конкретизирующих понятий.

Пусть выражением знания для опорного понятия  $C_i$  базовой компетенции ВК, будет следующее выражение:  $C_i \leq *C_{i1} (*C_1 + C_2) \sim + C_{ij} + C_{ik}$ ;

в котором учитывается изменчивость понятия  $C_i$  с помощью альтернативного необязательного понятия -  $C_{ij}$  и необязательного  $C_{ik}$ .

Понятийный граф выражения знания показан на рисунке 2.11:

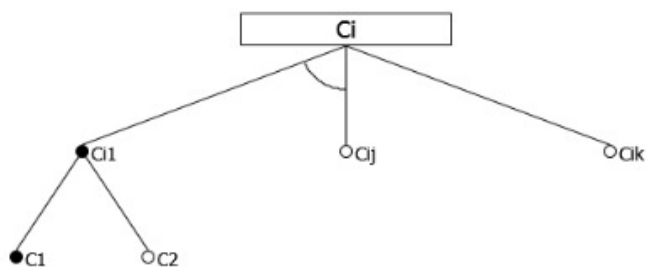


Рисунок 2.11 - Понятийный граф выражения знания

И, наконец, рассмотрим выражение знания опорного понятия  $C_i$  дополнительной компетенции ДК:  $C_i \leq *C_{i1} (+C_1 \sim +C_2 \sim +C_3) + C_{ij} \sim +C_{ik}$ ; в котором изменчивость понятия  $C_i$  осуществляется путем включения соответствующих необязательных и альтернативных понятий. Понятийный граф выражения знания показан на рисунок 2.12:

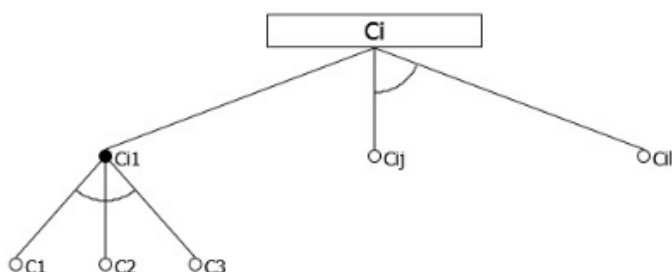


Рисунок 2.12 - Понятийный граф выражения знания

Как было показано, компетентностная модель позволяет моделировать, организовывать и структурировать знания в виде иерархии опорных и конкретизирующих понятий, а также концептуально выделять семантику отношений между понятиями компетентностной модели с помощью отношений общности и изменчивости.

### Выводы по 2 разделу

В данном разделе диссертации представлена методика формирования образовательных (знаниевых) компонент планируемого обучения - CDIO Syllabus. Представлен анализ подходов к классификации компетенций специалиста в форме представления «дерева» и «графа» компетенций. Конфигурирование требуемой образовательной траектории обучающегося сопряжено с формированием требований к его профессиональным компетенциям, представлением семантических знаний компетентностных моделей каждого из этапов инициативы CDIO, построением онтологий опорных понятий и их спецификации выражениями знаний.

### **3. АДАПТИВНАЯ ОБРАЗОВАТЕЛЬНАЯ СРЕДА**

Проект, названный инициативой CDIO, получил широкое распространение и охватывает образовательные программы в области техники и технологий. Задачей проекта является предоставление студентам обучения в основе которого лежит освоение инженерной деятельности, связанной с разработкой реальных систем, процессов и продуктов, в соответствии с моделью «Задумка - Проектирование - Реализация - Управление». В процессе обучения студенты должны получить богатый опыт ведения проектно-конструкторской и экспериментальной деятельности.

Начальным организационным моментом проекта является распределение студентов на команды исполнителей, в соответствии с целями, структурой и развитием бизнеса (предметной области). Понимание бизнеса увеличивает вероятность удовлетворения потребностей, связанных с достижением цели. От этого зависит также структура системы и терминология (понятия) для описания проекта. Кроме того, зная основные стороны бизнеса, можно разрабатывать легко развиваемые структуры.

Основные концепции и механизмы адаптивной образовательной среды нацелены на построение и использование компонент многократного использования, в роли которых выступают образовательные (знаниевые) компоненты, как результат онтологического инжиниринга предметной области.

Онтологический инжиниринг - это дисциплина, которая занимается разработкой и реализацией порождающих доменных моделей знаний - онтологий понятий, на базе которых формируются образовательные (знаниевые) компоненты планируемого обучения.

#### **3.1 Задачи адаптивной образовательной среды**

Адаптивная образовательная среда (АЕЕ- Adaptable Educational Environment) предназначена для формирования образовательных (знаниевых) компонент планируемого обучения CDIO Syllabus. В качестве предметной области среды рассматривается некоторая планируемая дисциплина специальности, в нашем случае такой дисциплиной выбрана дисциплина "Технология разработки распределенных приложений" специальности "Вычислительная техника и программное обеспечение, направления - "Инженерия программного обеспечения". Для данной дисциплины будет выполняться студенческий проект. В качестве основных положений таких дисциплин введем следующие правила:

1) Студенты выбирают проект и роль своего участия в проекте по следующим направлениям:

- инженерия предметной области и спецификация концептуальных моделей системы;
- прототипирование и дизайн интерфейса системы;
- архитектура и дизайн системы;
- модели и дизайн базы данных;

- спецификация сценариев, алгоритмизация и программирование;
- обеспечение качества программного продукта;
- техническое описание и документирование артефактов системы.

2) Работа по проекту должна происходить с использованием методов экстремального программирования (Agile) и с учетом принципа синергии процессного подхода, то есть с мозговой атакой в целом над проектом и в рамках своих задач и обязанностей, с планированием и сменой ролей своего участия над проектом. Принцип синергии процессного подхода - "Общая заинтересованность в конечном результате", обязателен для всех участников команды проекта.

3) Разработка проекта должна происходить в соответствии с этапами инициативы CDIO, то есть согласно этапов: Задумка, Проектирование, Реализация и Управление.

4) Каждый из этапов инициативы CDIO, в рамках профессионального профиля специальности, должен содержать компетентностную модель этапа, включающую семантические знания, необходимые для успешного выполнения бизнес-процессов и работ по студенческому проекту.

5) Компетентностная модель представляет собой композицию профессиональных, базовых и дополнительных компетенций, каждая из которых организуется и представляется онтологиями опорных понятий и соответствующими им конкретизирующими понятиями первого и второго уровней.

6) Компетентностные модели соответствующих этапов инициативы CDIO служат для формирования базовых и профильных модулей рабочего учебного плана профессионального профиля обучения. При этом, базовые и профильные модули должны состоять из множества профилирующих дисциплин, определяющих образовательный тренд рабочего учебного плана специальности.

7) Сформированный таким образом образовательный тренд из базовых и профилирующих дисциплин должен обеспечить формирование профессионального уровня знаний и умений, необходимого для успешного выполнения проектов и получения требуемой квалификации специальности.

За период обучения студенты должны выполнить, по крайней мере, два проекта. Первый проект выполняется на втором и третьем курсах, на которых планируются семестровые этапы инициативы CDIO. Например, этап Задумка - в 3 семестре, этап Проектирование - в 4 семестре, этап Реализация - в 5 семестре и этап Управление - в 6 семестре. Второй проект - дипломный проект, на котором все этапы инициативы CDIO должны быть реализованы в 7 и 8 семестрах.

### **Задача 1. Формирование образовательных (знаниевых) компонент планируемого обучения**

Профессиональный профиль специальности - это один из профилей специальности, для которого будут формироваться образовательные (знаниевые) компоненты планируемого обучения CDIO Syllabus. Таким

профессиональным профилем специальности "Вычислительная техника и программное обеспечение" примем "Системы реального времени, параллельных и распределенных приложений".

Студенческий проект. Предполагаются два студенческих проекта, соответственно на втором и третьем курсах, и дипломный проект - на четвертом курсе.

CDIO включает четыре этапа выполнения проекта - *Conceiving, Designing, Implementing* и *Operating*, для каждого из которых будет формироваться компетентностная модель этапа, являющаяся композицией профессиональных, базовых и дополнительных компетенций.

Компетентностная модель каждого из этапов CDIO состоит из:

- набора опорных понятий, определяющих семантические знания, необходимые для успешного выполнения бизнес-процессов и работ, предусмотренных этапом;
- онтологий и соответствующих выражений знаний, описывающих профессиональные компетенции этапа;
- онтологий и соответствующих выражений знаний, описывающих базовые компетенции этапа;
- онтологий и соответствующих выражений знаний, описывающих дополнительные компетенции этапа.

Структура онтологии опорного понятия представляет собой результат декомпозиции опорного понятия в виде иерархии идентифицирующих его понятий первого и второго уровней.

За основу отображения знаний в концептуальной базе данных образовательной среды принята алгебраическая модель, представляемая в форме выражения знания, элементами которого являются понятия, каждое из которых наделено свойством общности или изменчивости. Все входящие в выражение знания понятия являются ссылками на свое информационное содержание.

Таким образом, онтология опорного понятия и соответствующее выражение знания и представляют собой образовательный (знаниевый) компонент планируемого обучения - *CDIO Syllabus*, совокупность которых используется для последующего проектирования дисциплин специальности.

## **Задача 2. Проектирование дисциплин специальности**

Сформированные образовательные (знаниевые) компоненты планируемого обучения *CDIO Syllabus* должны использоваться для проектирования дисциплин учебного плана специальности.

Формированием и мониторингом образовательных (знаниевых) компонент планируемого обучения занимается эксперт (рисунок Б-1), а проектированием своих дисциплин - преподаватель.

Онтологии и их представления выражениями знаний хранятся в концептуальной базе данных образовательной среды и являются теми необходимыми знаниями, которые нужны преподавателю для проектирования

соответствующих дисциплин профессионального профиля учебного плана специальности, а обучающимся - для разработки своих проектов.

При проектировании дисциплины преподаватель должен определить, к каким модулям учебного плана относится проектируемая дисциплина - базовым, либо профильным и, если дисциплина относится к модулю базовых, тогда используются онтологии и выражения знаний базовых и дополнительных компетенций компетентностных моделей соответствующих этапов CDIO, а если проектируется профилирующая дисциплина - онтологии и выражения знаний профессиональных компетенций компетентностных моделей этапов CDIO.

Все, таким образом спроектированные на основе CDIO Syllabus дисциплины, будут составлять образовательный тренд профессионального профиля специальности, и вместе с другими дополнительными дисциплинами - рабочий учебный план специальности.

### **Задача 3. Структурирование учебного материала дисциплины для проведения занятий**

С помощью адаптивной образовательной среды и с использованием инструментальных средств среды Mind Map преподаватель представляет структуру учебного материала дисциплины, включающую:

- наименование дисциплины;
- перечень тем дисциплины;
- роль дисциплины в формировании профессиональных компетенций и квалификации студента;
- тема дисциплины, цель и назначение темы, представление учебного материала темы в традиционном формате;
- тема дисциплины и представление учебного материала темы в формате онтологий и выражений знаний;
- заключение и выводы по теме в терминах понятий онтологий и выражений знаний;
- список источников информации по изучаемой теме.

Инструментальное средство Mind Map используется: для структурирования учебного материала и его визуального представления в виде ориентированного графа, а также для управления процессом обучения, при этом все узлы ориентированного графа являются ссылками на соответствующую учебную и справочную информацию.

### **Задача 4. Адаптация знаний на основе онтологии**

Для мониторинга и адаптации знаний профессионального профиля специальности среда должна включать поисковую систему, которая осуществляла бы оперативный поиск и доставку необходимой информации из сети.

Поисковая система образовательной среды используя структуру информации, сформированную на основе онтологии опорного понятия, должна обеспечивать направленный поиск в рекомендуемых источниках полезной информации с описанием новых технологий и инструментальных средств по

специальности "Вычислительная техника и программное обеспечение" направления "Инженерия программного обеспечения", и смежных ИТ-областей.

Полученная таким образом информация представляется эксперту, на основании которой в концептуальную базу данных среды будут вводиться новые, либо корректироваться опорные понятия и их конкретизации по соответствующим профессиональным, базовым и дополнительным компетенциям компетентностных моделей каждого из этапов CDIO.

Большинство поисковых систем ищут информацию на сайтах Всемирной паутины, но существуют также системы, способные искать файлы на Ftp-серверах, товары в Internet-магазинах, а также информацию в группах новостей Usenet.

В последнее время появился новый тип поисковых движков, основанных на технологии RSS - семейство XML-форматов, предназначенных для описания лент новостей, анонсов статей, изменений в блогах и т.п.

Можно назвать и аналогичные технологии:

- Rich Site Summary (стандарт RSS 0.9x) - обогащенная сводка сайта;
- RDF Site Summary (RSS 0.9 и 1.0) - сводка сайта с применением инфраструктуры описания ресурсов;
- Really Simple Syndication (RSS 2.x) - очень простое приобретение информации.

Информация из различных источников, представленная в формате XML на базе RSS-стандартов, может быть собрана, обработана и представлена пользователю в удобном для него виде специальными программами-агрегаторами.

Таким образом, информационный поиск - это комплексная деятельность по сбору, организации, поиску, извлечению и распространению информации при помощи компьютерных технологий.

Примерами задач в области информационного поиска являются:

- информационный поиск документов по запросу пользователя;
- автоматическая рубрикация документов по заранее заданному рубрикатору;
- автоматическая кластеризация документов - разбиение на кластеры близких по смыслу документов;
- разработка вопросно-ответных систем - систем поиска точного фрагмента текста, отвечающего на вопрос пользователя, а не всего документа;
- автоматическое составление аннотации документа.

Единственный способ выразить информационную потребность пользователя состоит в такой форме изложения, которая была бы доступна для компьютерной обработки. Такая форма есть единственный способ связи между пользователем и поисковой машиной.

Поисковая система использует запрос как входные данные для получения того или иного результата - выборки из коллекции документов, соответствующих (релевантных) запросу. Пользователь оценивает результат

поиска по релевантности (мере соответствия) результата поиска и своей информационной потребности.

Перед началом работы поисковой системы следует индексировать коллекцию документов, путем создания индексных таблиц, которые позволяют значительно ускорить обработку запросов. Примером индексной таблицы является алфавитный или предметный указатель или оглавление книги.

Вообще индекс содержит все термины (понятия), которые появляются в документах коллекции, и такой индекс называется инвертированным словарем. Часто термины представляются в инвертированных файлах в виде токенов, то есть частей терминов, окончаний в терминах.

Для каждого токена в коллекции документов вычисляются следующие характеристики:

- число документов, в которых появился (встретился) данный токен;
- частота встречаемости токена в коллекции документов.

При извлечении релевантных документов важен способ обработки запроса.

Существуют следующие способы обработки запросов и вычисления релевантности для пары "запрос-документ":

**1. Булевский поиск** - в котором термины (понятия) запроса соединяются между собой с помощью логических операций *and*, *or*, *not* и группировкой терминов при помощи скобок. Такой запрос представляется логической формулой, в которой операндами являются термины, либо дополнительные условия.

В результате анализа информационных ресурсов интернета, поисковая система возвращает множество документов для которого логическая формула принимает истинное значение, или другими словами, каждому термину формулы сопоставляется множество документов, в которых данный термин встречается. Далее, над множествами документов выполняются элементарные операции объединения, пересечения и дополнения в результате которых получаем некоторое множество документов для которого логическая формула будет истинна.

Если  $D_0$  - множество всех документов коллекции, то  $D_T$  - множество документов в которых встречается термин  $T$ . Полезная выборка обозримого размера может быть получена, если задавать сложную логическую формулу.

**2. Ранжированный поиск** - основной способ обработки запросов поисковыми системами в интернете. Он основан на вычислении релевантности через распределение частот встречаемости терминов запроса по документам коллекции. Термины подвергаются сокращению для получения токенов, а также удаляются все стоп - слова ("где", "почему", и т.п.).

**3. Семантический поиск** - решается с помощью онтологии, а сам поиск осуществляется с учетом семантики терминов запроса и контекста, в котором эти термины используются. Благодаря онтологиям, появилась возможность создания семантической сети, которая должна стать надстройкой над WWW. В дополнение к поисковым движкам поисковые системы дополнены



специальными логическими движками, предназначенными для поиска и обработки информации в семантической сети.

В основе семантической сети лежат принципы: агрегация, безопасность и логика, а основную роль в работе с информацией играют интеллектуальные программные агенты.

Функциями интеллектуальных программных агентов являются нахождение источников информации, запрос данных, сопоставление и проверка этих данных на соответствие критериям поиска, и, наконец, выдача ответа в удобной для пользователя форме. Агенты руководствуются правилами логического вывода, используемыми в онтологиях, и алгоритмами нахождения оптимальных решений или уточнений условий запросов. Цепочки логических рассуждений, осуществляемые программными агентами, позволят получать информацию, представленную в интернет, в разрозненном виде, то есть из разных информационных источников.

### **3.2 Проектные решения задач образовательной среды**

Выше мы кратко описали четыре задачи образовательной среды, для каждой из которых должны быть приняты принципиальные проектные решения, во многом определяющие как сам процесс работы среды, так и качество и трудоемкость разработки. К таким решениям относят:

- выбор архитектуры программного обеспечения;
- выбор типа пользовательского интерфейса для администратора, эксперта и преподавателя и технологии работы с документами;
- выбор инструментов разработки и среды программирования.

Особенно важно определиться с представлением проекта, как такового, через призму типовых решений процесса - USDP; либо решений, связанных с процессом конфигурирования компонент OpenSource; либо решений, связанных с применением методов технологии Agile.

#### **3.2.1 Формирование образовательных (знаниевых) компонент планируемого обучения**

Решения эксперта связаны с:

- определением проекта и разбивкой выполнения проекта по этапам инициативы CDIO;
- определением перечня базовых и профилирующих дисциплин профессионального профиля учебного плана специальности;
- формированием образовательных (знаниевых) компонент, необходимых для проектирования базовых и профилирующих дисциплин профессионального профиля учебного плана специальности;
- проведением анализа внешних информационных ресурсов профессионального профиля специальности, с целью внесения изменений в образовательные (знаниевые) компоненты среды;
- осуществлением мониторинга и сопровождения работы среды.

Каждая из планируемых дисциплин профессионального профиля специальности должна иметь свой дескриптор, включающий следующие данные по дисциплине:

- название учебной дисциплины и ее трудоемкость в кредитах;
- ID дисциплины;
- название и шифр модуля, к которому принадлежит дисциплина (модуль - это как правило обязательный модуль по специальности, включающий циклы базовых и профильных дисциплин, составляющих основу специальности и направленные на формирование профессиональных компетенций. Как правило, один модуль состоит из одной или нескольких дисциплин);
- цели и задачи дисциплины;
- краткое описание содержания изучаемой дисциплины;
- список литературы, электронных учебников и источников сетевых образовательных ресурсов по дисциплине.

На рисунке 3.1 представлены системные компоненты среды, в нотации языка UML.

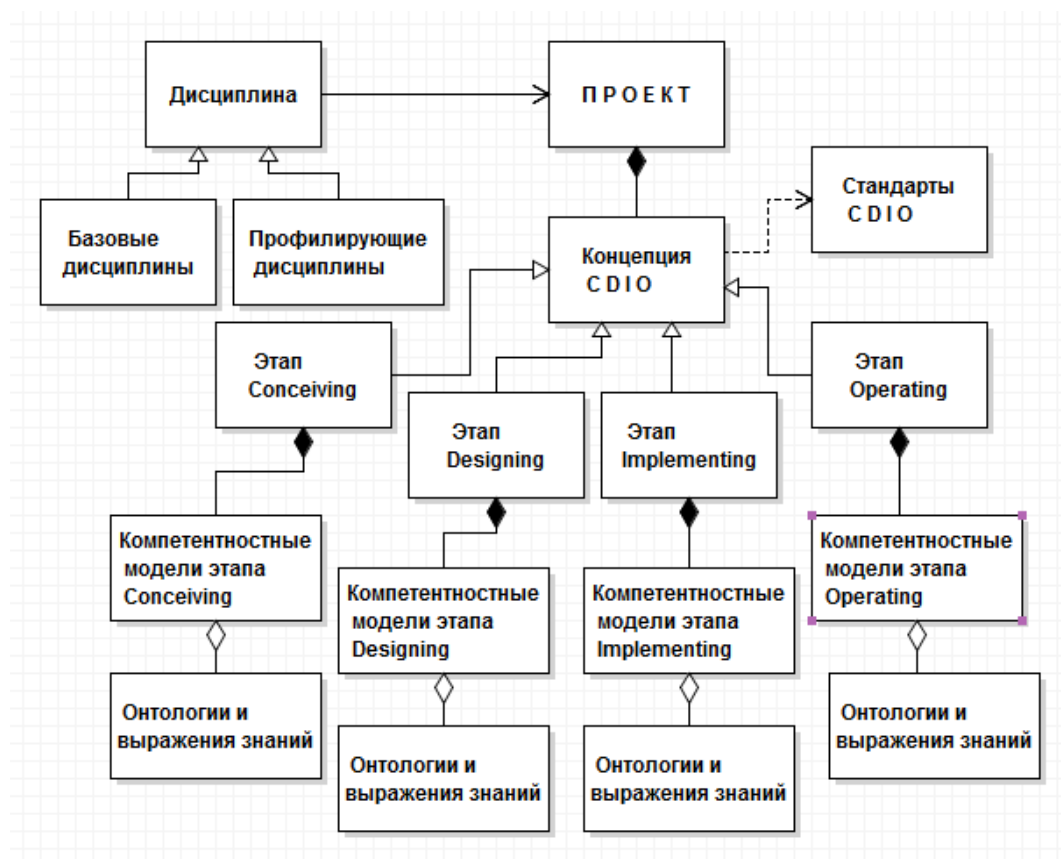


Рисунок 3.1- Системные компоненты среды

Основной страницей среды является страница эксперта, с помощью которой он выполняет следующие функции:

- 1) Формирует список проектов для студентов.
- 2) Выбирает этап CDIO по разработке проекта.

3) Описывает ДКМэ - дескриптор компетентностной модели этапа и формирует КМэ - концептуальную модель этапа CDIO являющуюся композицией профессиональных РК.КМэ, базовых ВК.КМэ и дополнительных DK.КМэ компетенций этапа.

4) Специфицирует опорные понятия профессиональных, базовых и дополнительных компетенций выражениями знаний и вводит их в концептуальную базу данных, тем самым формируя базу из образовательных (знаниевых) компонент для проектирования дисциплин учебного плана специальности.

Для отображения знаний в среде принята алгебраическая модель, которая имеет удобный формат для хранения, компьютерной обработки и визуализации. Для визуализации онтологий опорных понятий разработан парсер, либо можно воспользоваться инструментальным средством - Cytoscape.js.

Итоговым результатом работы эксперта является формирование двух файлов.

Первый файл - KC.Syllabus.Prof, содержащий онтологии и выражения знаний профессиональных компетенций всех этапов инициативы CDIO, предназначенный для проектирования профилирующих дисциплин специальности.

Второй файл - KC.Syllabus.Base, содержащий онтологии и выражения знаний базовых и дополнительных компетенций всех этапов инициативы CDIO и предназначенный для проектирования базовых дисциплин специальности.

Содержимое этих двух файлов преподаватель должен использовать для проектирования своих дисциплин.

5) И, наконец, последняя функция эксперта - работа с поисковой системой для проведения мониторинга внешних информационных ресурсов, редактированием базы знаний и внесением изменений, связанных с появлением новых технологий и средств как по направлению "Инженерия программного обеспечения", так и по смежным областям.

На рисунке 3.2 представлена главная страница среды для работы эксперта.

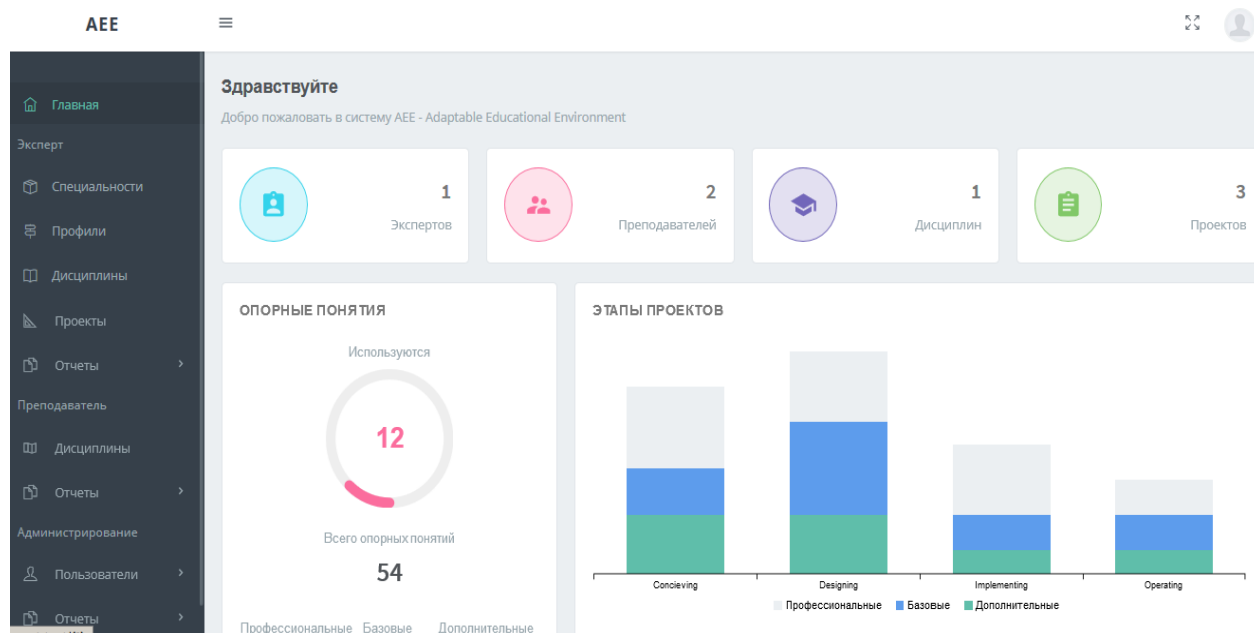


Рисунок 3.2- Главная страница интерфейса эксперта

Дизайн главного интерфейсного окна эксперта соответствует требованиям по переходу на проектный подход в обучении. То есть в основу планирования учебного плана и самого процесса обучения должен лежать проектный подход при котором студенты проходят этапы обучения также, как это происходит при разработке программных и информационных систем, включающих этапы анализа, проектирования, реализации и сопровождения.

На рисунке 3.3 приведена диаграмма вариантов использования, отражающая модульную архитектуру среды. Для каждого из модулей определен сценарий выполнения потоков событий, связанных с выполнением соответствующих функций эксперта.

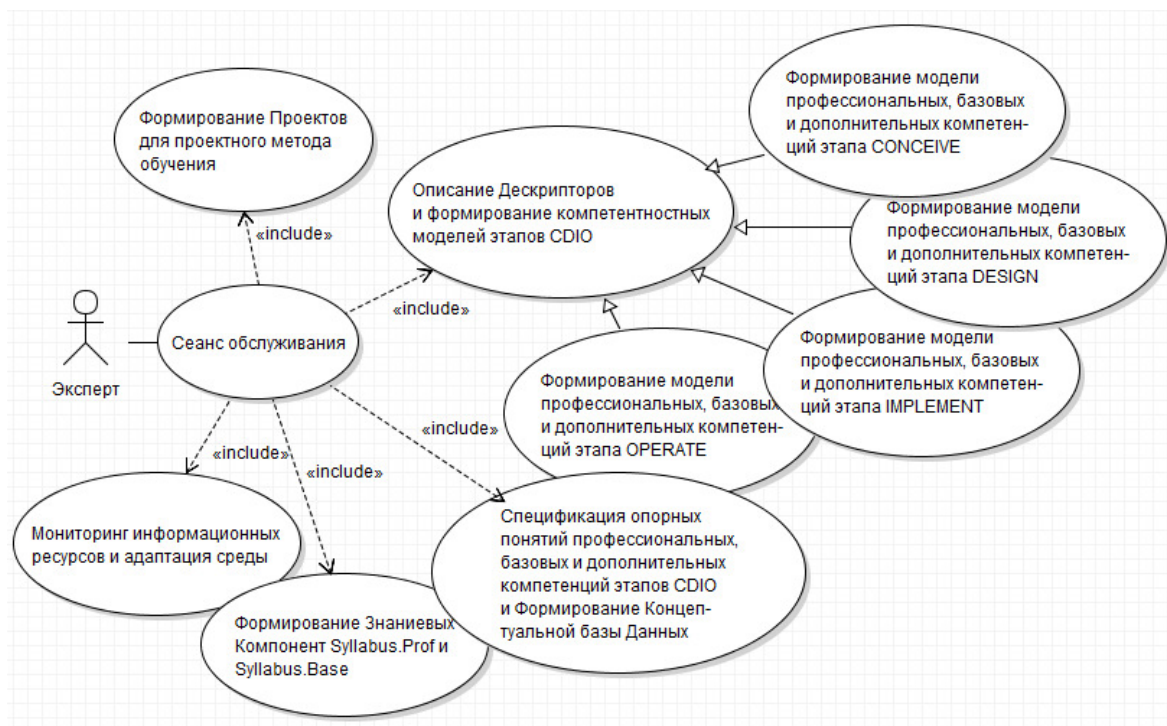


Рисунок 3.3- Диаграмма вариантов использования адаптивной образовательной среды

### 3.2.2 Проектирование дисциплин специальности

Дисциплины учебного плана специальности могут принадлежать одному из циклов: циклу общеобразовательных дисциплин, либо циклу базовых дисциплин, либо циклу профилирующих дисциплин специальности. В этих циклах дисциплина может быть обязательной компонентой для изучения, либо быть компонентом по выбору.

Приведем ряд определений, сформулированных в Правилах организации учебного процесса по кредитной технологии обучения [21].

Индивидуальный учебный план студента (ИУПС) - план, формируемый на каждый учебный год на основании типового учебного плана и каталога элективных дисциплин. ИУПС определяет индивидуальную образовательную траекторию каждого обучающегося отдельно.

Рабочий учебный план (РУП) - документ, разрабатываемый на основе типового учебного плана специальности и индивидуальных учебных планов студентов.

Элективные дисциплины - учебные дисциплины, входящие в компонент по выбору в рамках установленных кредитов, отражающие индивидуальную подготовку студента, учитывающие специфику социально-экономического развития и потребности конкретного региона и сложившиеся научные школы высшего учебного заведения.

Каталог элективных дисциплин- представляет собой систематизированный аннотированный перечень всех дисциплин компонента по выбору, содержащий их краткое описание с указанием цели изучения, краткого содержания

основных разделов и ожидаемых результатов изучения, то есть приобретаемых обучающимися знаний, умений, навыков и компетенций.

Модульная образовательная программа.

Модули подразделяются на следующие виды:

1) Тип 1 - общие обязательные модули - циклы базовых дисциплин, формирующие общие компетенции и напрямую не связанные со специальностью;

2) Тип 2 - обязательные модули по специальности - циклы базовых и профилирующих дисциплин, составляющие основу специальности и направленные на формирование профессиональных компетенций;

3) Тип 3 - модули по выбору для определенной специальности - циклы дисциплин по индивидуальному профилированию, направленные на формирование возможных компетенций в рамках специальности;

4) Тип 4 - модули по выбору, выходящие за рамки квалификации - циклы дисциплин, не относящиеся к специальности и направленные на формирование дополнительных компетенций (информационные технологии, иностранные языки);

5) В случае, если модуль объединяет несколько взаимосвязанных между собой дисциплин, одна из которых является пререквизитом другой, то изучение этих дисциплин должно быть последовательным. В данном случае эти дисциплины изучаются в разных семестрах;

6) Если модуль состоит из смежных дисциплин, составные компоненты которого не являются пререквизитом других составных компонентов, то возможно параллельное изучение таких смежных дисциплин модуля;

7) Результаты обучения по модулю формулируются в виде предметных и межпредметных компетенций;

8) Как правило, модуль формируется из:

- больших по объему (три и более кредитов) и завершенных по времени и содержанию дисциплин;

- небольшого объема и взаимно дополняющих друг друга дисциплин;

- нескольких родственных, смежных дисциплин, которые обеспечивают преемственность учебной программы и представляют различные уровни одной дисциплины.

9) Смежные дисциплины обеспечивают междисциплинарность модуля и направлены на формирование широкого спектра необходимых компетенций, выходящих за рамки одного модуля.

Дескриптор дисциплины (CD - Course Description) - информация о дисциплине, которую Эксперт планирует для включения в рабочий учебный план специальности, на основании сформированных образовательных (знаниевых) компонент планируемого обучения CDIO Syllabus специальности.

Спецификация дескриптора дисциплины

Концепция общности инженерного подхода к описанию дисциплин проявляется в том, что SWEBOOK никак не затрагивает конкретных аспектов, связанных, например, с языками программирования, реляционными базами

данных, сетевыми технологиями и т.п., а предлагает руководство к своду знаний, которыми специалист должен владеть вне зависимости от неминуемых перемен в конкретных технологиях и инструментальных средствах. Поэтому разработчик, владея багажом знаний определенных в SWEBOOK, должен выбрать технологическое решение соответствующее его задаче, наилучшим образом.

Применение концепции изменчивости осуществляется с помощью характеристической модели и таких ее элементов, как: диаграмма характеристик; определение характеристик; правила декомпозиции характеристик и логическое обоснование характеристик [52]. Такие характеристики составляют содержание дескриптора дисциплины специальности по которым можно будет четко разграничивать и определять каждую из дисциплин, вносить в дисциплину соответствующие изменения, дополнения и т.п.

Для обеспечения гарантированного уровня качества обучения содержание учебной дисциплины должно:

- носить деятельностный характер;
- соответствовать современному уровню научно-технического прогресса;
- носить опережающий характер;
- носить проблемный характер;
- носить интегрированный характер;
- содержание должно включать современные интеллектуальные технологии;
- содержание должно переводиться с эмпирического уровня на концептуальный.

В связи с отмеченным, дескриптор  $i$ -й дисциплины  $CD_i$  должен включать в себя следующие характеристики:

1. ID дисциплины -  $f_{i1}$ ;
2. Название дисциплины -  $f_{i2}$ ;
3. Трудоемкость дисциплины (в кредитах, где кредит - унифицированная единица измерения объема учебной работы) -  $f_{i3}$ ;
4. Описание модуля дисциплины -  $f_{i4}$ . Данная характеристика должна содержать следующие компоненты: название модуля и его шифр; тип модуля; количество кредитов; семестр; пререквизиты модуля; краткое содержание модуля; результаты обучения (формулируются в виде предметных и межпредметных компетенций); продолжительность модуля (один или два семестра); литература по дисциплинам модуля.
5. Описание дисциплины -  $f_{i5}$ . Данная характеристика должна содержать следующие компоненты: цели-  $f_1$ , задачи-  $f_2$  и содержание -  $f_3$  дисциплины. Во всех компонентах описания дисциплины Эксперт должен раскрыть приобретаемые обучающимися знания, умения, навыки и компетенции, которые данная дисциплина вносит в компетентностную модель выпускника;
6. Инфраструктура дисциплины -  $f_{i6}$ , включающая  $f_1$  - пререквизиты дисциплины, содержащие знания, умения и навыки, необходимые для освоения

изучаемой дисциплины и  $f_2$  - постреквизиты дисциплины, для изучения которых требуются знания, умения и навыки, приобретаемые по завершении изучения данной дисциплины.

7. Источники информации -  $f_{i7}$ :  $f_1$  - список литературы,  $f_2$  - список электронных учебников,  $f_3$  - ссылки на сетевые образовательные ресурсы.

Приведем несколько примеров описания дескриптора дисциплины с помощью характеристической модели.

Пусть имеем описание дескриптора  $D_i$ -й дисциплины учебного плана по всем указанным семи характеристикам, что можно представить следующим выражением спецификации:

$$CD_i \leq *f_{i1} * f_{i2} * f_{i3} * f_{i4} * f_{i5} (*f_1 * f_2 * f_3) * f_{i6} (*f_1 * f_2) * f_{i7} (*f_1 + f_2 * f_3); \quad (3.1)$$

и визуализировать соответствующим характеристическим графом - рисунок 3.4:

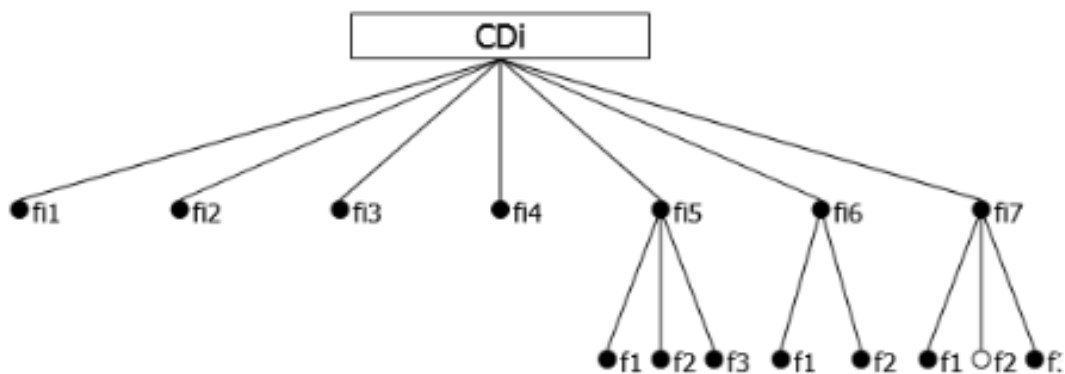


Рисунок 3.4- Характеристический граф выражения

Приведем пример простой спецификации дескриптора дисциплины, включающего два обязательных понятия -  $C_{i1}$ ,  $C_{i2}$ , три обязательных характеристик -  $f_{i1}$ ,  $f_{i2}$ ,  $f_{i4}$  и одну альтернативную необязательную характеристику -  $f_{i3}$ , причем, обязательная характеристика  $f_{i2}$  идентифицируется двумя характеристиками второго уровня -  $f_1, f_2$ :

$$CD_i \leq *C_{i1} * C_{i2} * f_{i1} * f_{i2} (*f_1 + f_2) \sim +f_{i3} * f_{i4}; \quad (3.2)$$

соответствующий граф изображен на рисунке 3.5:



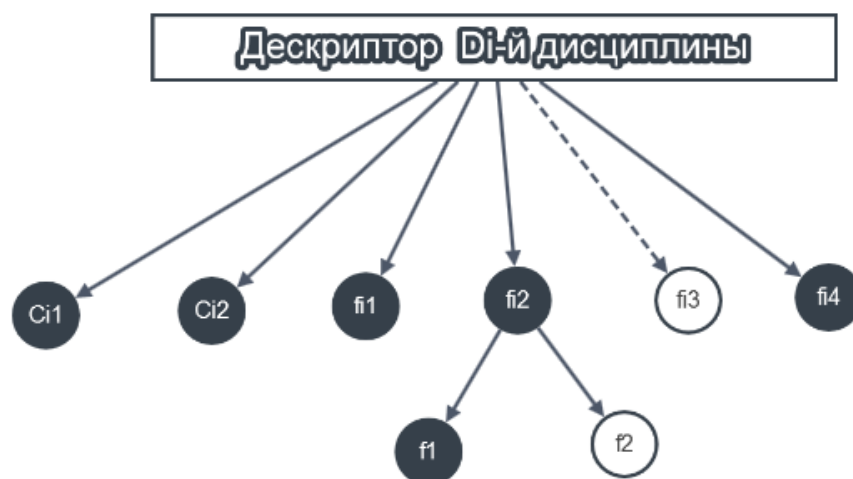


Рисунок 3.5 - Смешанный граф выражения

Таким образом, в данном разделе диссертации было введено понятие - "Дескриптор дисциплины" (Course Description - CD), включающее метаданные дисциплины. Для спецификации свойств дисциплины использованы выражение спецификации и характеристическая модель, с помощью которых можно отображать обязательные, необязательные и альтернативные характеристики.

Формированием дисциплин профессионального профиля специальности занимается эксперт, как ответственное лицо с большим педагогическим, научным и профессиональным опытом работы в областях, связанных с инженерией программного обеспечения (Рисунок Б.4).

Таким образом, главная учебно-методическая задача эксперта заключается в формировании дескрипторов дисциплин учебного плана и разработке соответствующих образовательных (знаниевых) компонент этих дисциплин.

Пример дескриптора дисциплины "Технология разработки распределенных приложений"

Каждая из дисциплин адаптивной образовательной среды должна иметь свой дескриптор, включающий следующие характеристики дисциплины:

- название учебной дисциплины и ее трудоемкость в кредитах;
- ID дисциплины;
- название и шифр модуля, к которому принадлежит дисциплина (модуль - это как правило обязательный модуль по специальности, включающий циклы базовых и профильных дисциплин, составляющих основу специальности и направленные на формирование профессиональных компетенций. Как правило, один модуль состоит из одной или нескольких дисциплин);
- цели и задачи дисциплины;
- краткое описание содержания изучаемой дисциплины;
- список литературы, электронных учебников и источников сетевых образовательных ресурсов по дисциплине.

Приведем пример дескриптора дисциплины "Технология разработки распределенных приложений" профессионального профиля "Системы

реального времени, параллельных и распределенных приложений" специальности "Вычислительная техника и программное обеспечение", направления - "Инженерия программного обеспечения".

1. Название учебной дисциплины и ее трудоемкость в кредитах - "Технология разработки распределенных приложений", трудоемкость - три кредита.

2. ID дисциплины - TDS

3. Название и шифр модуля, к которому принадлежит дисциплина - "Параллельные системы реального времени", шифр - CSRT.

4. Цели и задачи дисциплины - Технологический прогресс в производительности микропроцессоров сделали рентабельными распределенные системы и системы реального времени на базе микрокомпьютеров. В большинстве научных и учебных изданиях, посвященных объектно-ориентированной концепции анализа и проектирования программного обеспечения, рассматриваются только последовательные приложения и опускаются существенные вопросы, связанные с проектированием систем реального времени, параллельных и распределенных систем.

Объединение концепций объектно-ориентированного проектирования с концепциями параллельного выполнения необходимо для успешного создания распределенных приложений, работающих в реальном масштабе времени. В данной дисциплине, помимо стандартной нотации UML, особое внимание уделяется моделированию динамики системы, представляющему интерес для приложений реального времени и распределенных приложений.

Дисциплина "Технология разработки распределенных приложений" рассматривает объектно-ориентированный анализ и проектирование параллельных систем с использованием нотации UML, в частности, рассматриваются важнейшие категории параллельных приложений: распределенные и реального времени.

Цель дисциплины заключается в том, чтобы научить обучающихся строить аналитические и проектные модели и оценивать модели и артефакты параллельных и распределенных систем и систем реального времени.

Задачами дисциплины являются:

– освоение различных концепций, технологий, жизненных циклов и методов проектирования систем реального времени, параллельных и распределенных систем;

– освоение метода COMET для разработки параллельных приложений и этапов моделирования требований, аналитического моделирования и проектного моделирования;

– освоение способов расчета производительности проекта параллельной системы реального времени.

5. Краткое описание содержания изучаемой дисциплины.

В данной дисциплине рассматриваются критерии разбиения, призванные помочь программисту выявлять подсистемы, объекты и параллельно

выполняемые задачи на разных стадиях процессов анализа и проектирования. С помощью динамического моделирования взаимодействия объектов и конечных автоматов излагаются способы совместного использования диаграмм кооперации и состояний. Во всех построениях и моделях делается акцент на параллельности для чего приводятся характеристики активных и пассивных объектов. Рассматриваются проектирование распределенных приложений и методы взаимодействия распределенных компонентов. С помощью теории планирования в реальном времени анализируется производительность системы реального времени. Приводится развернутый пример приложения "Банковская система", иллюстрирующий применение изложенных концепций и методов.

### **3.2.3 Структурирование учебного материала дисциплины**

Применение визуальных моделей и, как следствие, визуального мышления для обучения требует от преподавателя рассматривать следующие этапы структурирования учебной информации.

Первый этап - проведение анализа структуры и организации контента учебного материала дисциплины, таким образом, чтобы нацелить студентов на активное и продуктивное восприятие, так называемый подход "от студента на предмет", в отличие от традиционного подхода "от предмета на студента".

Второй этап - создание мыслительных образов, получаемых путем ассоциативного связывания понятий темы дисциплины и смежных с ней дисциплин, что предполагает определенные усилия студентов по формированию целостной системы знаний, отвечающей поставленной учебной задаче как отдельного занятия, так и курса в целом.

Третий этап - закрепление учебного материала, путем приобретения навыков и умений, и который по своим целям и учебным возможностям можно отнести к поисковой деятельности. Этот этап требует включения различных механизмов, поведение которых обусловлено взаимодействием понятий учебного материала, структурно определенных и связанных между собой.

Как правило, учебная программа позволяет преподавателю варьировать объем и последовательность содержания в зависимости от конкретной цели. Цели изучения материала соответствуют уровням его усвоения - это может быть опознание, воспроизведение, специфицирование и конструирование. Между понятиями и структурными элементами содержания учебного материала должны быть намечены ассоциативные связи.

Применительно к профессиональному обучению особенно важно учитывать цели обучения, в соответствии с будущей профессиональной деятельностью обучаемых. С позиций этой деятельности и должен рассматриваться вопрос о существенности тех или иных ассоциативных связей и в целом последовательности изучения учебного материала.

Разрабатываемая структура и организация контента учебного материала обычно представлена в различных методических документах. Самыми простыми и распространенными формами являются традиционно текстовое изложение и его план. Полный текст изложения однозначно определяет ее

структуру, но недостаточно обозрим, не дает о ней наглядного представления и, следовательно, не позволяет оценить ее оптимальность. План контента учебного материала более обозрим, отражает принятую структуру, но не содержит деталей и структурных связей между элементами учебного материала.

Гораздо более эффективно отображать содержание учебного материала визуально. Для этого используют такие наглядные формы как графы, семантические сети, спецификации учебных элементов, конспект - схемы, а также средства и технологии современных информационных технологий. Наряду с визуализацией рекомендуется составлять спецификацию учебных элементов, опорный конспект и другие применяемые формы и приемы профессиональной деятельности педагога. Преподаватель предварительно структурирует учебную информацию со всеми ее связями и отношениями оставляя для себя спецификации, графы и прочие «строгие» формы, обучающимся же можно предложить более «образные» визуальные материалы.

Структурирование содержания учебной информации, на наш взгляд, должно начинаться с разработки онтологий, то есть с выделения основных учебных элементов, так называемых опорных понятий дисциплины, их идентифицирующих понятий, установления ассоциативных связей между ними и построения иерархии понятий - онтологий.

Учебный элемент - это подлежащая усвоению логически законченная часть учебной информации, являющийся носителем собственной информации, отсутствующей в других учебных элементах. Структурно учебный элемент является неделимой частью учебной информации. Естественно, что неделимость учебного элемента понятие условное и в другом случае при более подробном рассмотрении может детализироваться, и наоборот.

В зависимости от конкретного содержания учебной информации в качестве учебного элемента могут быть: определение понятия, факт, явление, процесс, закономерность, принцип, способ действия, характеристика объекта, вывод или следствие, при этом следует иметь в виду, что способ выражения понятия (модель, выражение или граф) не является учебным элементом, как таковым.

Структура контента учебного материала создается всей совокупностью учебных элементов, включенных в определенные связи [74]. Можно выделить следующие типы связей: взаимодействие, порождение, преобразование, строение, управление и функциональные связи. К выделенным типам связей следует добавить свойства общности и изменчивости.

Часто связь сама выступает как учебный элемент, то есть как информация, подлежащая усвоению. Подобное имеется в объектном моделировании, когда ассоциация между двумя классами выступает в качестве ассоциированного класса, имеющего совместные свойства и поведение ассоциированных классов.

Для составления спецификации учебных элементов необходимо провести структурно-логический анализ содержания учебных элементов, то есть выделить учебные элементы и установить отношения между ними. Выделенные

учебные элементы дифференцируются на опорные и конкретизирующие понятия, причем на основе опорных понятий можно формировать новые знания и приемы умственной и практической деятельности.

Для визуального отображения структуры и содержания учебного материала используется среда построения когнитивных карт - Mind Map.

Первый узел Map-карты - дисциплина, при клике на который появляется следующая информация о дисциплине: название дисциплины; роль и место дисциплины в профессиональной подготовке выпускника; место и связь дисциплины с другими дисциплинами учебного плана специальности.

Второй узел карты обозначает тему дисциплины, при выборе которой появляется информация: название темы; роль и значение темы в дисциплине; какие профессиональные компетенции специальности развивает данная тема.

Далее от узла темы дисциплины следует последовательность образовательных узлов:

1) учебный контент темы, при клике на который представляется традиционно полный текст изложения темы лекции.

2) модели, понятия и выражения знаний темы, при клике на который представляются модели онтологий, опорные понятия и их выражения знаний.

3) заключение и выводы по теме, при клике на который показывается семантическая суть темы с помощью онтологий опорных понятий и их спецификаций выражениями знаний.

4) контрольные вопросы и упражнения по теме, в которых используются опорные и конкретизирующие понятия.

5) список литература и интернет - источников информации по теме.

Каждый из описанных узлов сопровождается поясняющей информацией.

Дисциплина - это более или менее стабильная образовательная единица учебного плана, а составляющие ее темы могут подвергаться изменениям и адаптации.

Для визуального отображения содержания учебного материала и его понятийной структуры используем среду построения когнитивных карт - Mind Map.

Покажем последовательность фрагментов - этапов представления учебной информации при проведении лекционных занятий, на примере дисциплины "Программная инженерия".

Лекция начинается с объяснения преподавателем роли дисциплины в профессиональной подготовке, а также с местом и связью этой дисциплины с другими дисциплинами учебного плана специальности, как представлено на рисунке 3.6.

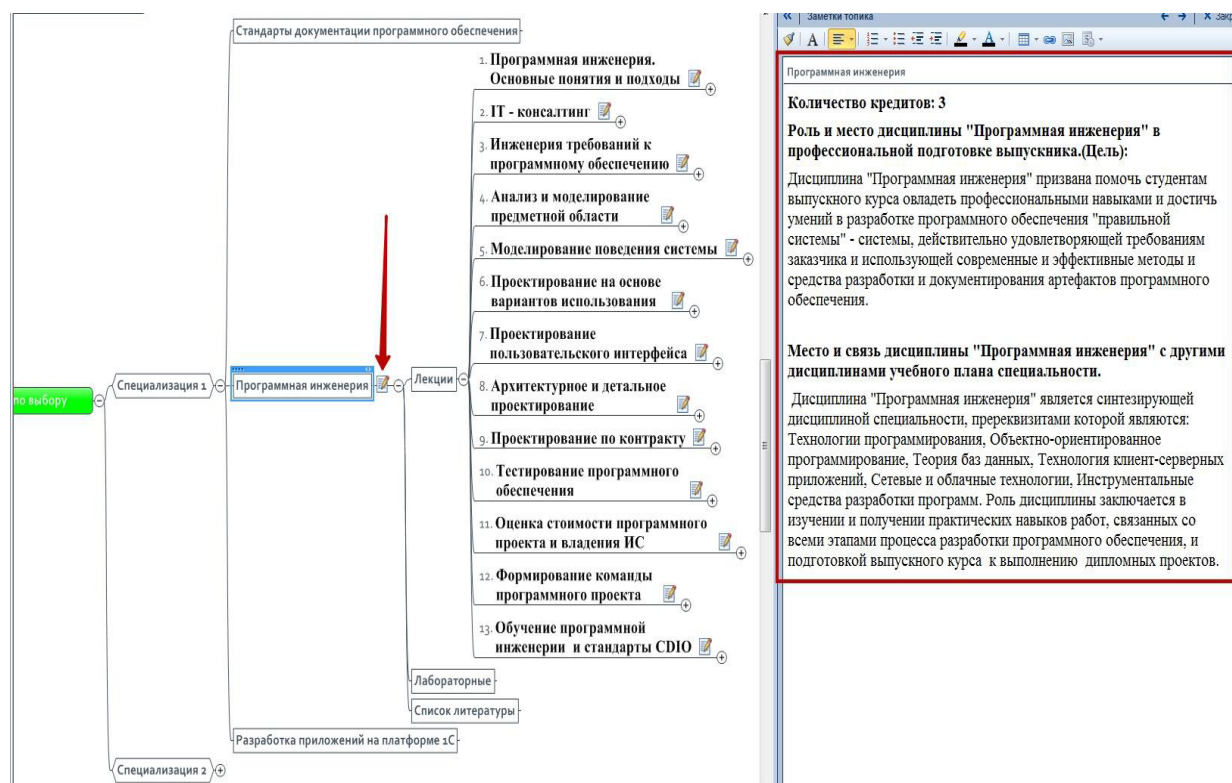


Рисунок 3.6 - Представление места и связи дисциплины с другими дисциплинами специальности

Далее, выбирается тема лекции по которой студентам разъясняется роль, которую играет эта тема в дисциплине и какие профессиональные компетенции специальности она развивает (рисунок 3.7).

Далее учебный контент темы представляется в обычном традиционном текстовом формате (рисунок 3.8), после изучения которого обучающиеся осмысливают данный материал и структурируют его путем выделения опорных понятий учебного контента темы и, по возможности, конкретизирующих понятий контента темы.

После краткого обсуждения возможных онтологий опорных понятий, преподаватель демонстрирует требуемые модели, понятия и выражения знаний учебного контента темы (рисунок 3.9), которые обсуждаются со студентами, с целью внесения соответствующих поправок в их представления.

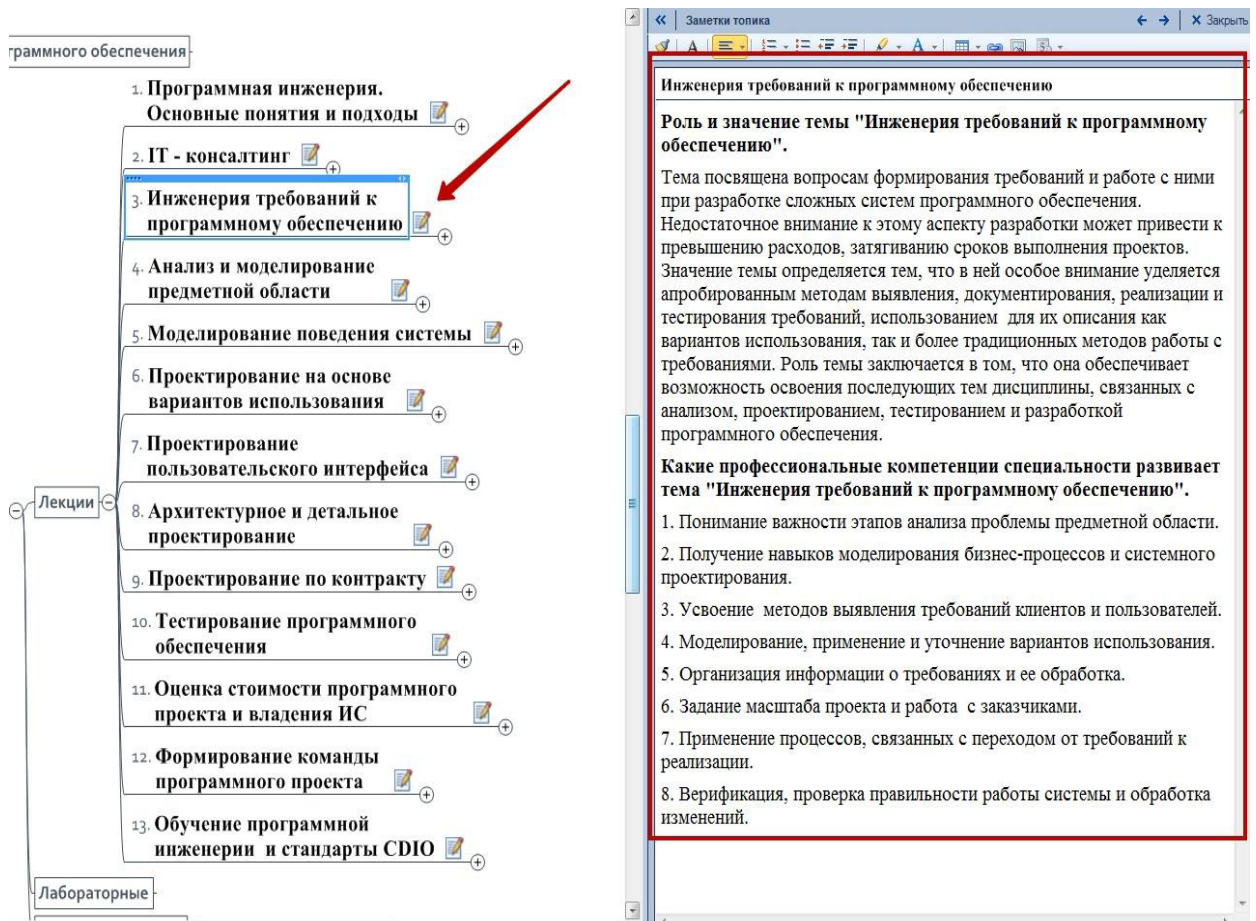


Рисунок 3.7 - Роль и значение темы дисциплины

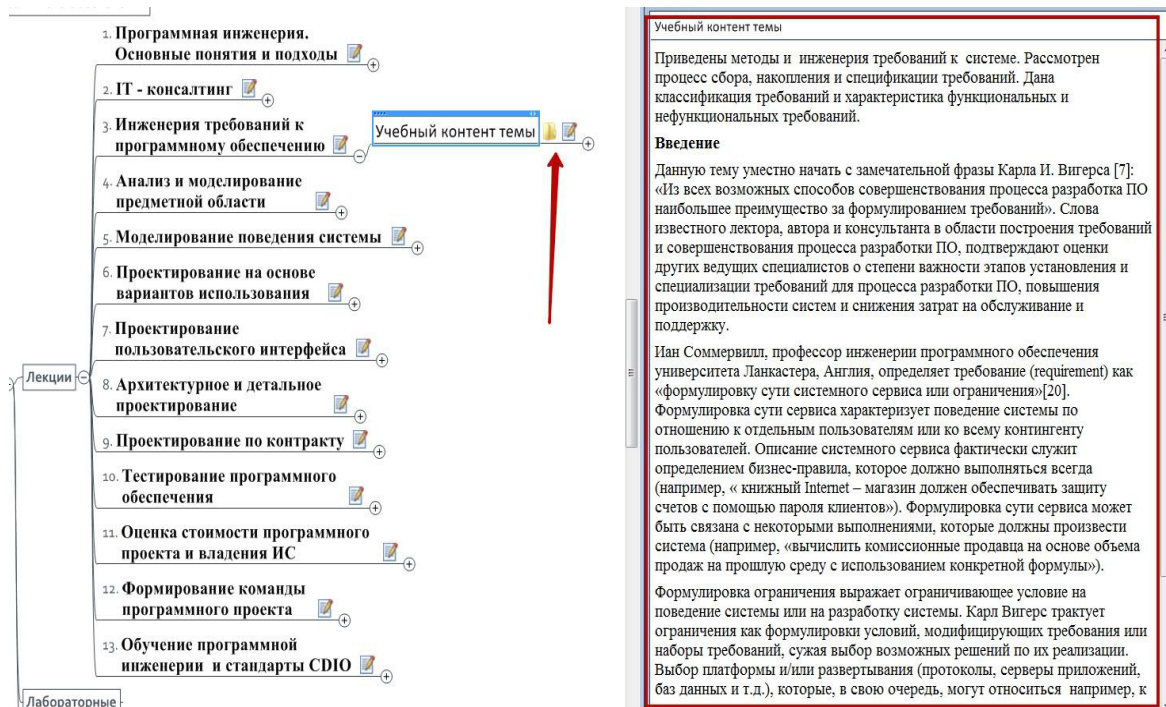


Рисунок 3.8 - Представление учебного контента темы в традиционном текстовом формате

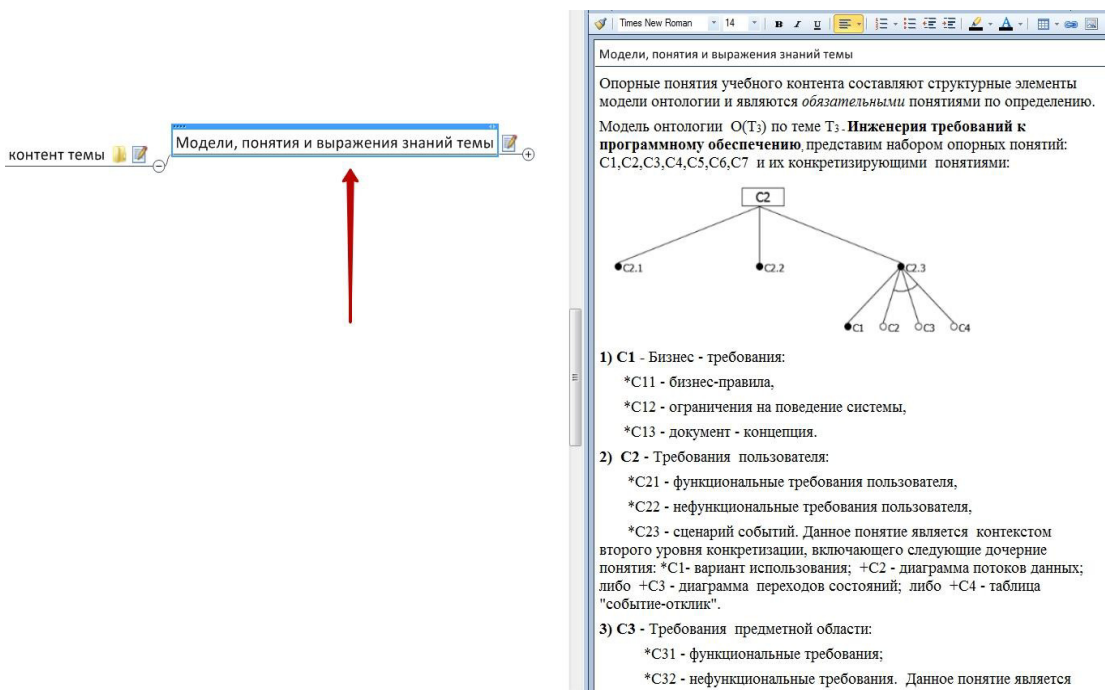


Рисунок 3.9 - Модели, понятия и выражения знаний учебного контента темы

На заключительных стадиях лекционного занятия преподаватель приводит краткое заключение по теме, используя при этом опорные понятия и их выражения знаний; проверяет уровень усвоения лекции и в заключении знакомит с источниками информации по теме, рисунки 3.10, 3.11, 3.12 соответственно.

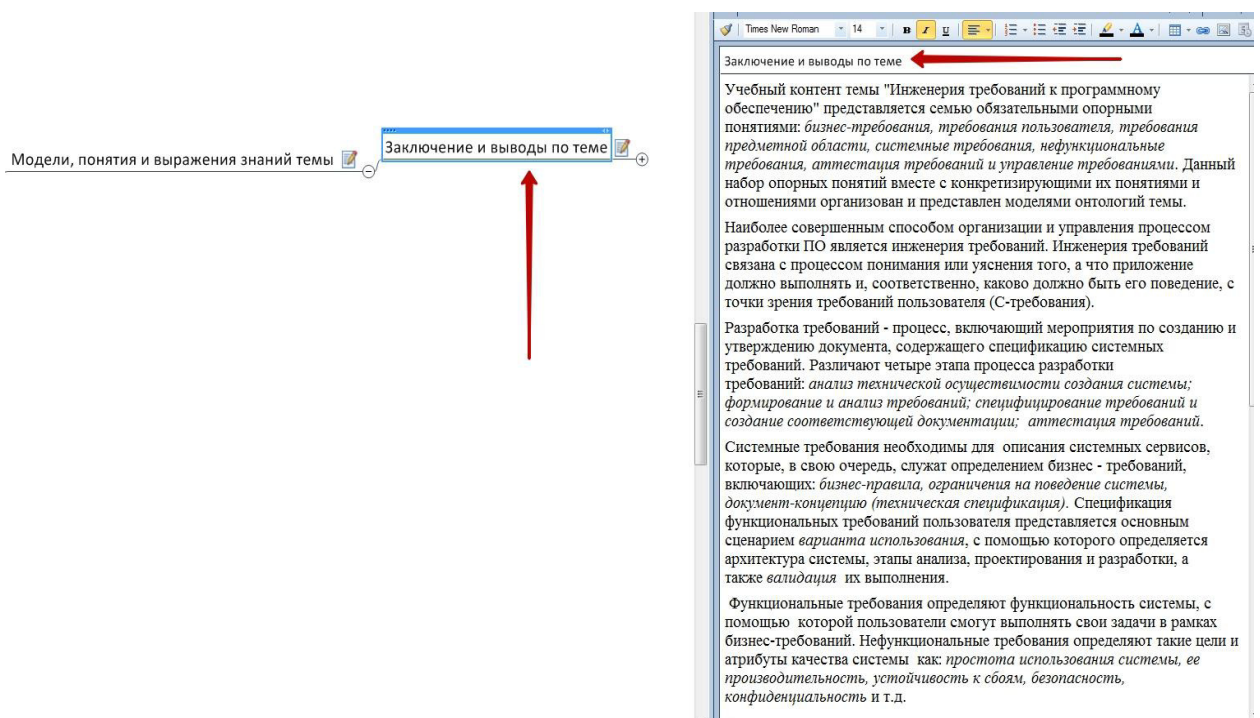


Рисунок 3.10- Заключение и выводы по теме, в соответствии со структурой понятий темы



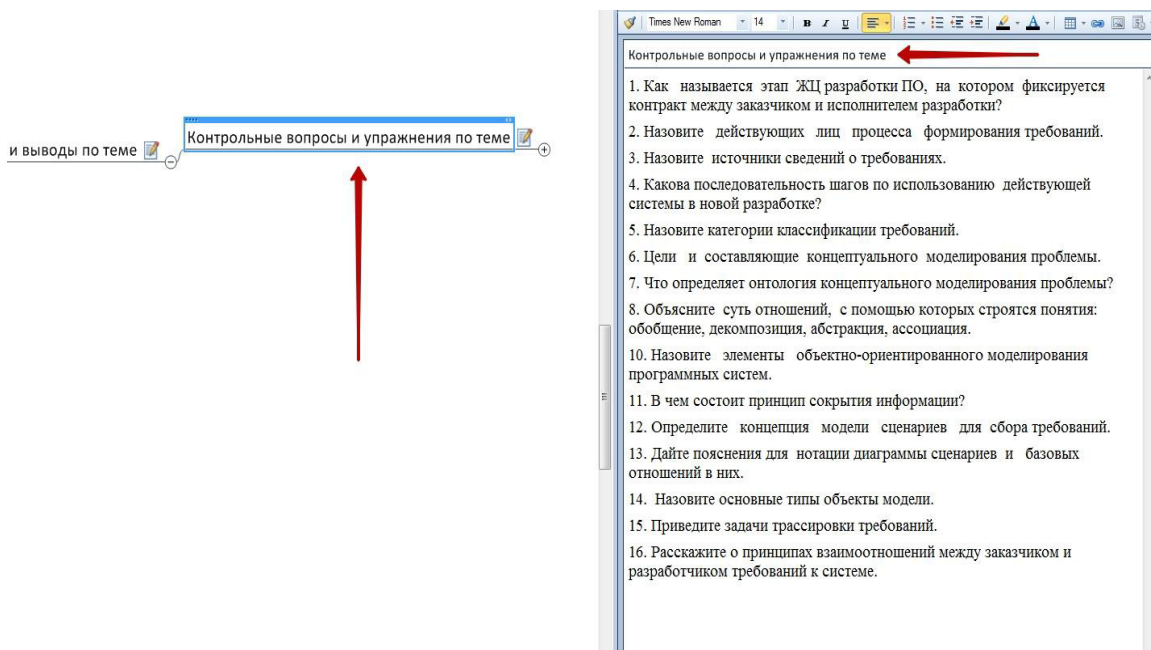


Рисунок 3.11- Контрольные вопросы и упражнения по теме

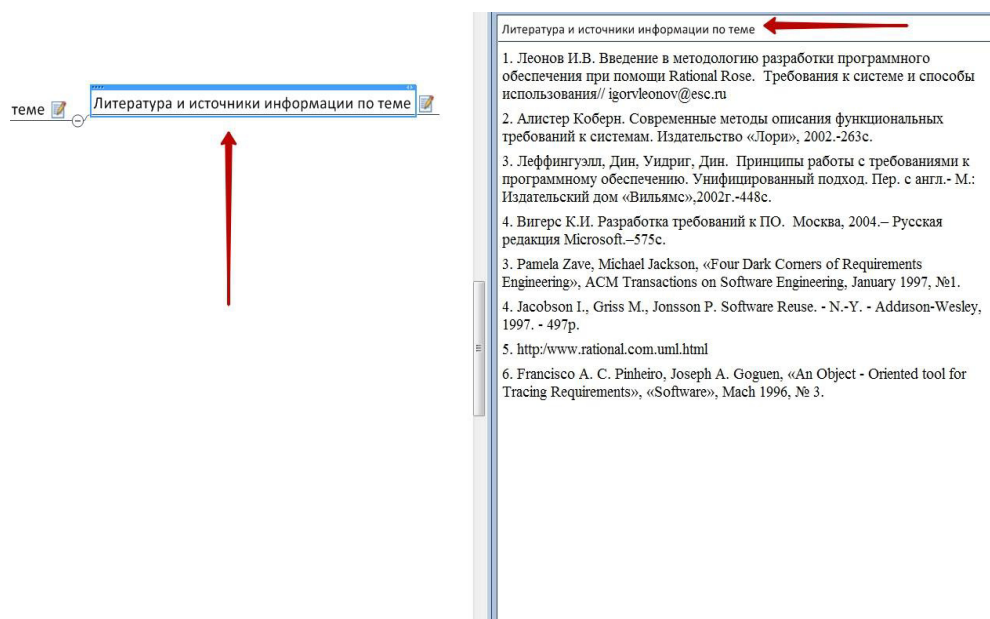


Рисунок 3.12- Литературные источники и дополнительная информация по теме

### Выводы по 3 разделу

В данном разделе рассмотрены основные концепции и механизмы адаптивной образовательной среды, направленные на построение и использование компонент многократного использования, в роли которых выступают образовательные (знаниевые) компоненты, как результат онтологического инжиниринга предметной области. Показаны последовательность фрагментов - этапов представления учебной информации при проведении лекционных занятий, на примере дисциплины "Программная инженерия"

## 4. ПОИСКОВАЯ СИСТЕМА ПРОЕКТНО – ОРИЕНТИРОВАННОЙ СРЕДЫ

### 4.1 Проектные решения, создание микросервисов

Целью разработки поисковой системы проектно-ориентированной образовательной среды является обеспечение автоматической доставки новых статей областей знаний, связанных с направлением "Инженерия программного обеспечения", а также различные информационные ресурсы, используя для этого технологию агрегации информации на микросервисной архитектуре.

Проектно-ориентированная образовательная среда позволит форсайтные исследования, прогнозирование научно-технических и инновационных проектов и образовательных программ, осуществляющиеся с учетом следующего минимального количества исходных данных:

- библиометрический и патентный обзор;
- информация о состоянии и тенденциях развития науки и технологий в мире и Казахстане;
- информация о технологических потребностях в образовании;
- имеющиеся отечественные научные разработки в рамках тематики дисциплины, компетенции научных организаций в данной отрасли;
- компетенций преподавательского состава.

Программирование микросервисов - небольших автономных компонентов, позволяет добиться модульности и отказоустойчивости любой программы. Теория микросервисов тесно связана с философией Unix, способствует улучшению архитектуры любых приложений, дает возможность избегать громоздкого и запутанного кода, масштабировать любые проекты и ювелирно разрабатывать даже самые сложные системы [72].

В данном разделе, приведем проектные решения, связанные с созданием микросервисов и, используя диаграммные техники и нотации языка объектного моделирования UML, опишем соответствующие артефакты программного обеспечения поисковой системы: диаграммы вариантов использования, последовательности, пригодности, объектов, компонентов, развертывания и классов.

Архитектуру поисковой системы составляют модули, связанные с агрегацией данных, извлечением данных из контекста, отображением данных пользователям и администрированием данных.

Агрегатор. Данный модуль периодически проводит опрос интернет - ресурсов на наличие новых данных и, в случае их обнаружения, сохраняет веб - страницу целиком в базе данных.

Экстрактор: Извлекает из веб - страницы главное изображение, текст, ссылку на первоисточник, дату и время публикации, а также дату и время импорта.

Единый API: Данный модуль работает как единая точка входа с помощью которой осуществляется поставка, сохранение и изменение данных. Режимы

сохранения и редактирования данных предусмотрены только для Эксперта среды.

Панель администрирования: Данный модуль является приложением для Администратора среды, которому предоставляются возможности удаления, редактирования и просмотра новостей.

Индексатор данных: Модуль осуществляет индексирование данных для быстрого полнотекстового поиска

Сервис кэширования данных: Данный модуль позволяет в фоновом режиме вносить данные по популярным запросам в кэш память.

В отличие от монолитных приложений, микросервисная архитектура гораздо более гибка в отношениях между микросервисами. В монолитных проектах каждый проект является узлом. Однако, в микросервисах каждый сервис потенциально является как узлом, так и хабом. То есть, во-первых, каждый микросервис может как вносить данные в другие микросервисы, так и получать данные из остальных сервисов; и, во-вторых, на основе каждого микросервиса могут работать остальные сервисы. Это даёт возможность использовать любой микросервис для обработки данных в системе.

Следует отметить сильные стороны и возможные недостатки каждого из двух типов архитектуры.

На рисунке 4.1 представлена монолитная архитектура приложения.

Монолитная архитектура означает, что приложение - это большой связанный модуль, где все компоненты спроектированы так, чтобы работать вместе с друг другом, имея при этом общую память и ресурсы. Основное преимущество монолитной архитектуры в том, что многие приложения обычно используют типовой функционал, связанный с логированием, предоставлением права доступа и т.д. Такая архитектура дает преимущество в производительности, так как общий доступ к памяти осуществляется быстрее, чем коммуникация между процессами (IPC- Interprocess Communication).

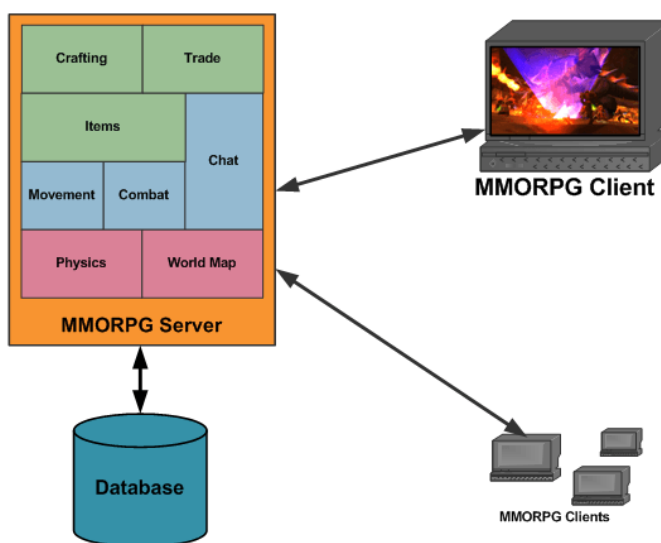


Рисунок 4.1 - Монолитная архитектура приложения

В процессе эволюции монолитные приложения становятся сильно связанными и запутанными, что усложняет процессы изолирования сервисов для независимого масштабирования и поддержки кода, а также, понимание кода контроллеров и сервисов.

Монолитный подход более выигрышен в плане производительности для проектов малой и средней сложности.

На рисунке 4.2 представлена микросервисная архитектура приложения.

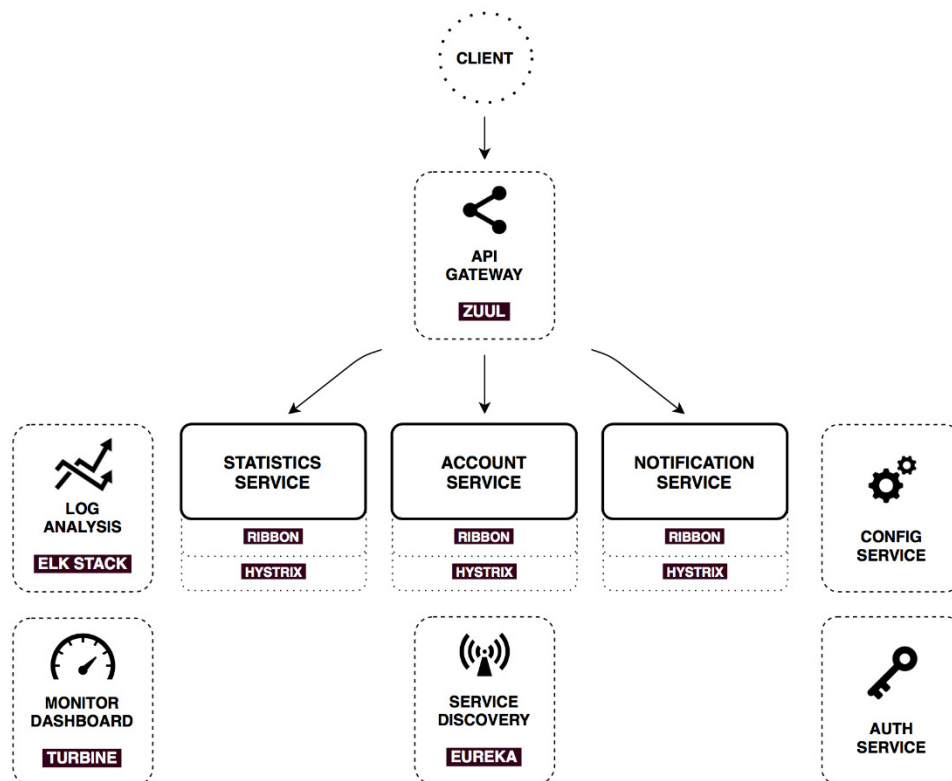


Рисунок 4.2 - Микросервисная архитектура приложения

Микросервисная архитектура означает, что приложение состоит из маленьких независимых приложений, работающих со своими собственными ресурсами и развивающимися независимо друг от друга, при этом, приложения потенциально могут быть размещены на разных машинах.

Микросервисы архитектурно лучше организованы, так как каждый сервис занимается определённой задачей и независим от работы других компонентов. Несвязанные сервисы легче адаптировать для обслуживания различных приложений. Также очевидны преимущества в производительности, в зависимости от того как они организованы, потому что микросервисная архитектура позволяет изолировать проблемные места и масштабировать их независимо от остального приложения.

Исходя из этого можно выделить следующие преимущества микросервисной архитектуры:

Гибкость и продуктивность. Команда, разрабатывающая микросервис в состоянии полностью понимать его кодовую базу, что позволяет осуществлять сборку, разворачивание и тестирование микросервиса независимо от других компонентов и в более быстром цикле разработки. Поскольку микросервис всего лишь другой компонент в компьютерной сети, поэтому его можно разрабатывать с использованием наиболее подходящего для реализуемой функциональности языка программирования, фреймворка и механизма долговременного хранения. Это гарантирует использование новых технологий или обновленных версий, без необходимости ожидать готовности других разработчиков.

Масштабируемость. Любой микросервис можно масштабировать во время исполнения, независимо от других микросервисов, тем самым обеспечивая наиболее эффективное использование ресурсов и оперативную реакцию на изменение нагрузки. Теоретически, нагрузка на компонент может быть перенесена на наиболее подходящую для решения задачи платформу. Данная миграция может быть выполнена полностью, независимо от остальных компонентов, с целью получить преимущество за счет правильного размещения в сети.

Устойчивость. Каждый микросервисный компонент должен быть разработан так, чтобы требования к собственной доступности выполнялись без необходимости выставлять данные требования к другим компонентам приложения.

## **4.2 Моделирование поисковой системы**

### **4.2.1 Варианты использования поисковой системы**

Предметная область - это совокупность знаний, значимая для пользователей и, соответственно область специализации или интереса, внутри которых находятся семейства целых систем.

Концептуальная модель предметной области – абстрактная модель, описывающая систему понятий предметной области.

К объектам, которые взаимодействуют в системе, относятся: информационные ресурсы, необработанные и обработанные статьи, пользователи и комментарии. Каждый из перечисленных объектов имеет свои свойства. Так свойствами информационного ресурса являются название ресурса и ссылка на первоисточник; свойствами для необработанной статьи являются - ссылка на ресурс, дата добавления, код страницы статьи из первоисточника; свойствами обработанной статьи являются - заголовок, текст, изображение, комментарии, категория; свойствами пользователя являются - почтовый адрес, имя пользователя и, наконец, свойствами комментария являются пользователь, статья и текст комментария.

Вариантом использования (англ. use case) в разработке программного обеспечения и системном проектировании является сценарий поведения объектов системы, при взаимодействии с внешней среды. Система может

отвечать на внешние запросы, либо сама может выступать инициатором взаимодействия. Другими словами, вариант использования описывает, «кто» и «что» может делать с системой, или что система может сделать с «кем» или «чем». Методика вариантов использования применяется для выявления пользовательских и функциональных требований к поведению системы.

В поисковой системе есть несколько вариантов использования, как отображено в таблице 4.1. Посетитель сайта - это пользователь, который просматривает статьи на сайте, выполняет поиск и оставляет комментарии.

Таблица 4.1 - Варианты использования поисковой системы

| Требование  | Действующее лицо | Вариант использования               |
|---|------------------|-------------------------------------|
| Пользователь попадает на главную страницу сайта       | Посетитель сайта | Отображение главной страницы        |
| Пользователь выбирает интересующую категорию          | Посетитель сайта | Выбор категории                     |
| Пользователь использует поиск для сокращения выдачи   | Посетитель сайта | Поиск информации по ключевому слову |
| Пользователь переходит на следующую страницу          | Посетитель сайта | Переход по страницам сайта          |
| Пользователь заходит на страницу статьи для прочтения | Посетитель сайта | Отображение полного текста статьи   |

#### 4.2.2 Диаграммы вариантов использования

Диаграмма вариантов использования описывает функциональное назначение системы или то, что система должна делать.

Необходимость разработки диаграммы вариантов использования обусловлена достижением следующих целей:

- определением общей границы и контекста моделируемой предметной области;
- формулированием общих требований к функциональному поведению проектируемой системы;
- разработкой исходной концептуальной модели системы для ее последующей детализации в форме логических и физических моделей;
- подготовкой исходной документации для взаимодействия разработчиков системы с ее пользователями.

Суть диаграммы вариантов использования состоит в следующем. Проектируемая система представляется в виде множества сущностей или актеров, взаимодействующих с системой с помощью вариантов использования. При этом актором (actor) или действующим лицом называется любая сущность, взаимодействующая с системой извне. Это может быть человек, техническое устройство, программа или любая другая система, которая может служить источником воздействия на моделируемую систему так, как определит сам разработчик. Вариант использования служит для описания сервисов, которые

система предоставляет актору. Диаграмма вариантов использования может дополняться пояснительным текстом, который раскрывает семантику составляющих ее компонентов.

Рассмотрим диаграмму вариантов использования пользователя сайта.

Основной актер на сайте это пользователь системы. Пользователи могут авторизоваться для комментирования статей и изменения личных данных. Но для просмотра статей размещенных на сайте авторизация не требуется, это могут делать все пользователи сайта независимо от уровня доступа. Также пользователь может сократить выдачу статей для удобства чтения, воспользовавшись поиском по ключевым словам или по разделу. Рассмотрим его возможности на диаграмме вариантов использования, как показано на рисунке 4.3.

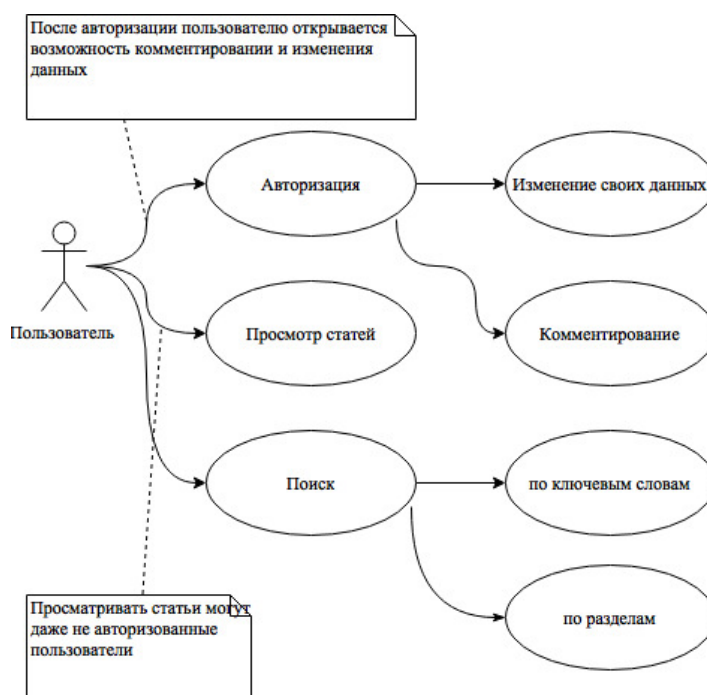


Рисунок 4.3 – Диаграмма вариантов использования пользователя сайта

Далее, рассмотрим диаграмму вариантов использования оператора ресурсов.

Оператор ресурсов отвечает за добавление новых ресурсов на сайт и поддержку уже существующих. Также оператор ресурсов может редактировать существующие статьи на сайте. Диаграмма вариантов использования оператора ресурсов показана на диаграмме 4.4.

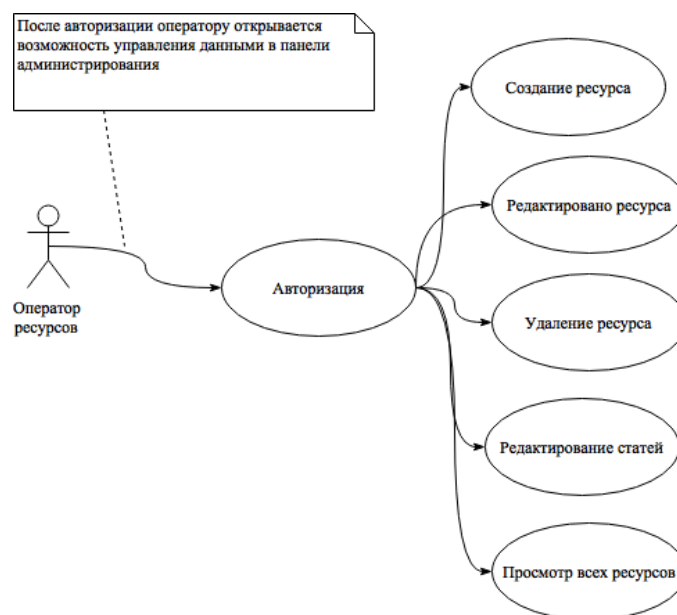


Рисунок 4.4 – Диаграмма вариантов использования оператора ресурсов

### 4.2.3 Моделирование динамических аспектов поисковой системы

Диаграмма последовательности является основным элементом детального проектирования, то есть динамического моделирования поведения объектов в виде упорядоченного по времени последовательности обмена сообщениями между объектами системы.

Для каждого варианта использования создается диаграмма последовательностей, в которой описываются главная и альтернативные последовательности событий над объектами системы. В результате получается ядро динамической модели, в котором определено поведение системы во время выполнения и то, как реализуется это поведение.

Диаграмма последовательностей состоит из четырех основных элементов:

1. текста последовательности действий варианта использования, который записывается сверху вниз с левой стороны от диаграммы;
2. объектов, представленных в виде прямоугольников, в которых записывается имя или номер экземпляра объекта и имя класса объекта;
3. сообщений, изображаемых стрелками, направленными от одного объекта к другому;
4. d) методов (операций) объектов, представляемых в виде узкого прямоугольника - фокуса управления; метод объекта владеет управлением системой вплоть до точки, в которой прямоугольник кончается.

Авторизация - одна из самых важных частей практически любого проекта, с помощью которой, пользователи проходят процедуру аутентификации и получают доступ к модулям поисковой системы.

Диаграмма последовательности варианта использования "Авторизация пользователя", показана на рисунке 4.5.



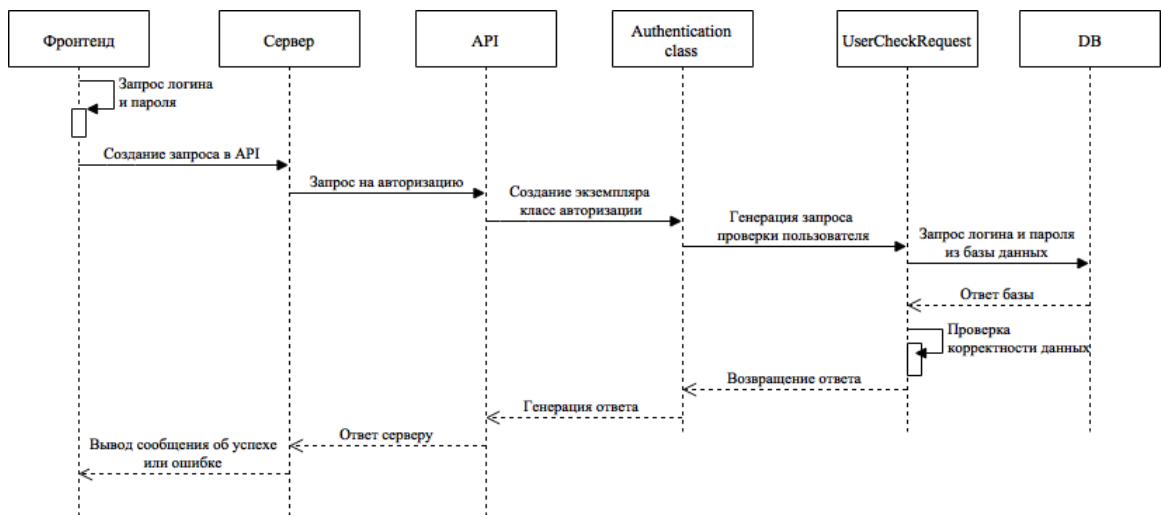


Рисунок 4.5 - Диаграмма последовательности варианта использования "Авторизация пользователя"

Пользователи могут оставлять на сайте свои комментарии к статьям. Работа комментирования изображена на следующей диаграмме последовательности варианта использования "Добавление комментария" (рисунок 4.6).

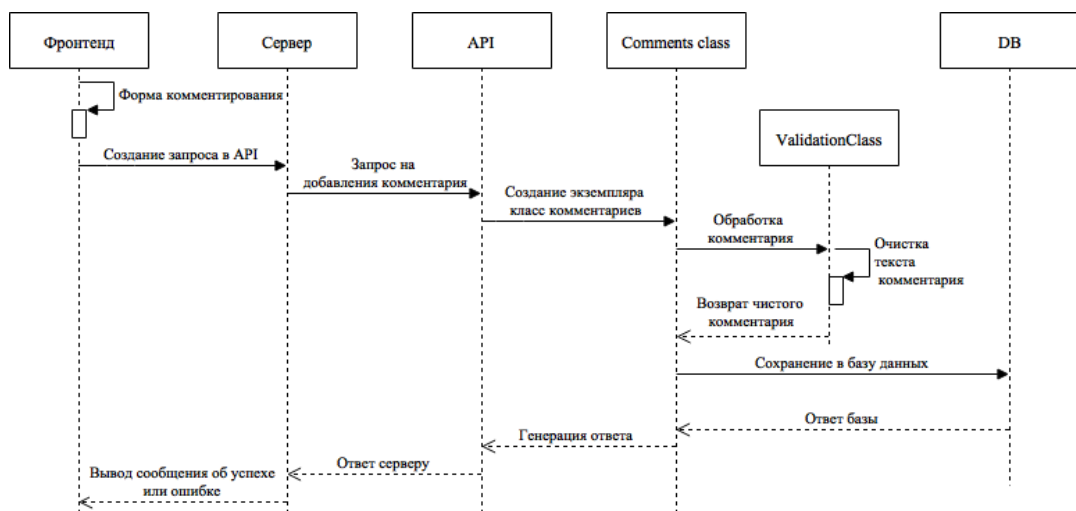


Рисунок 4.6 - Диаграмма последовательности варианта использования "Добавление комментария"

Как можно заметить, схема работы клиентской части системы с базой данных, похожа для всех запросов. Пользователь работает с фронтендом, при каких-либо действиях, запрос посылается на сервер, на котором генерируется запрос в единый API, который отвечает за бизнес-логику поисковой системы и выполняет все основные действия.

Front-end и back-end - термины в программной инженерии, с помощью которых, согласно принципу разделения ответственности, различают внешнее

представление от внутренней реализации соответственно. Front-end - интерфейс взаимодействия между пользователем и основной программно-аппаратной частью (back-end). Front-end и back-end могут быть распределены между одной или несколькими системами.

В архитектуре программного обеспечения может быть много уровней между аппаратной частью и конечным пользователем, каждый из которых также может иметь front-end и back-end.

Front end - это абстракция, которая предоставляет пользовательский интерфейс. Например, в проектировании программного обеспечения, Model-View-Controller архитектура обеспечивает front-end и back-end между базой данных, компонентами обработки данных и пользователями.

Рассмотрим пример работы одного из микросервиса - агрегатора, который работает независимо от всей системы и выполняется без помощи «актеров». Схема его работы предельно проста и изображена на рисунке 4.7.

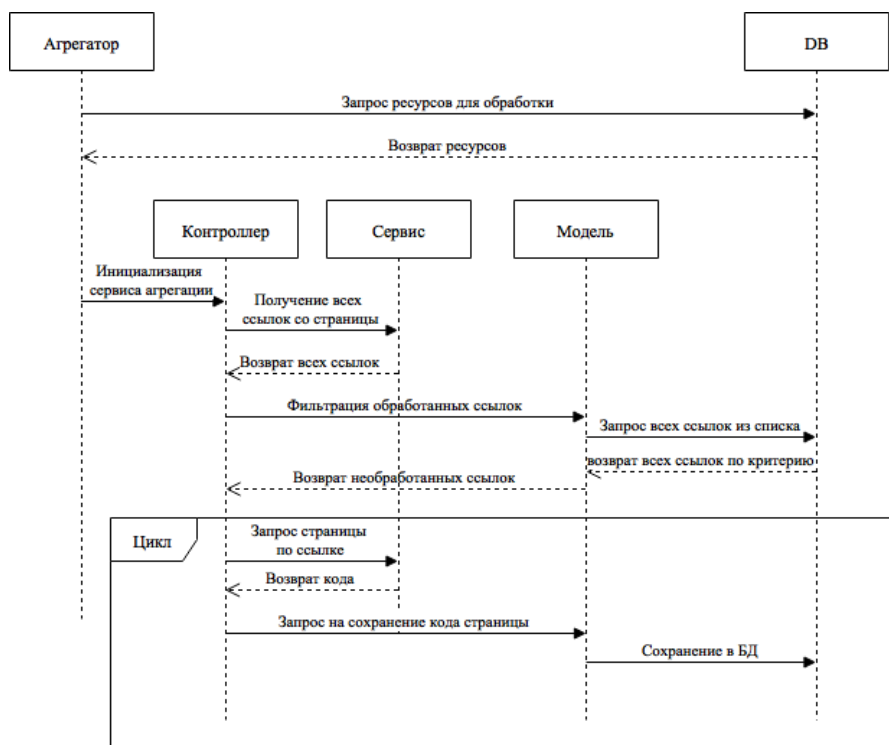


Рисунок 4.7 – Диаграмма последовательности «микросервис - агрегатор» поисковой системы

На рисунке 4.8 изображена схема работы экстрактора - сервиса, для извлечения текста статьи из кода страницы. Это ключевой микросервис, который отвечает за появление новых статей.

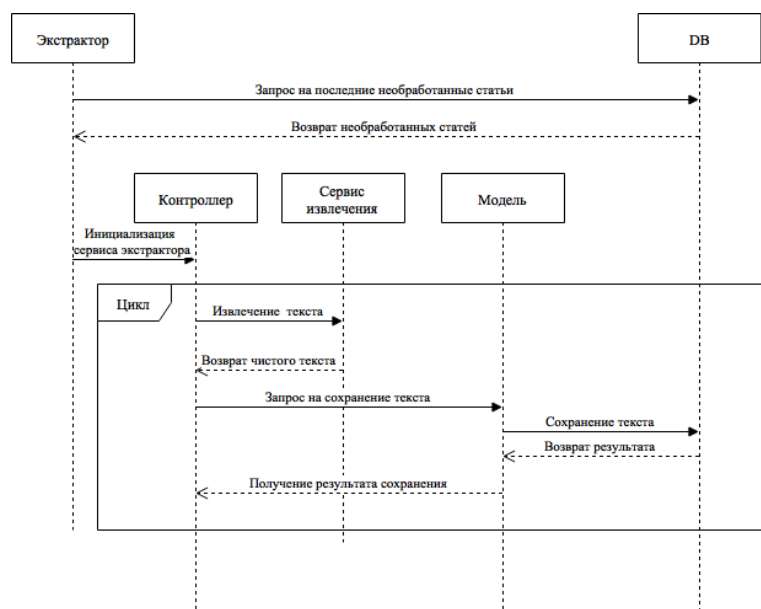


Рисунок 4.8 – Диаграмма последовательности экстрактора поисковой системы

### Выводы по 4 разделу

В данном разделе приведены проектные решения по созданию микросервисов, спецификация которых выполнена с помощью диаграммных техник и нотации языка объектного моделирования UML. Описаны следующие артефакты поисковой системы: диаграммы вариантов использования, последовательности, пригодности, объектов, компонентов, развертывания и классов, необходимые для проектирования и разработки поисковой системы, технологическая платформа которой основана на микросервисной архитектуре.

Архитектуру поисковой системы составляют модули, связанные с агрегацией данных, извлечением данных из контекста, отображением данных пользователям и администрированием данных.

Полученные решения позволят обеспечить автоматическую доставку новых статей пользователям проектно-ориентированной образовательной среды. Непосредственная реализация форсайтных исследований для определения приоритетных направлений, технологий и тематик научных исследований

Также рассмотрены вопросы, связанные с текущим положением ИТ-рынка по предоставлению информационных услуг; современными тенденциями и требованиями в сфере ИТ-образования и, в частности, по направлению "Инженерия программного обеспечения"; обоснована технологическая целесообразность и эффективность использования автоматической агрегации контента на микросервисной архитектуре.

Функции поисковой системы ориентированы на сбор данных с интернет-ресурсов; извлечение заголовков, текста и другой информации из кода веб-страницы; управление статьями; обеспечение прослойки кэша между пользователем и API; организации единой точки входа для редактирования и просмотра данных.

## 5 СЦЕНАРИИ РАБОТЫ ОБРАЗОВАТЕЛЬНОЙ СРЕДЫ

### 5.1 Сценарий работы эксперта

Адаптивная образовательная среда (Adaptable Educational Environment) исполнена в виде веб-приложения, предоставляющего возможность формирования образовательных (знаниевых) компонент планируемого обучения CDIO Syllabus, с целью проектирования дисциплин учебного плана специальности.

При проектировании и реализации среды были использованы следующие платформы, технологии и инструментальные средства разработки:

- интегрируемая среда разработки Visual Studio 2015, как набор инструментов для создания программного обеспечения от планирования до разработки пользовательского интерфейса, написания кода, тестирования и отладки, анализа качества кода и производительности, развертывания в средах клиентов;

- платформа ASP.NET MVC, в которой шаблон архитектуры Model-View-Controller (MVC) разделяет приложение на три основных компонента: модель, представление и контроллер. Платформа ASP.NET MVC представляет собой альтернативу схеме веб - форм ASP.NET при создании веб - приложений;

- в качестве СУБД среды использована система управления реляционными базами данных Microsoft SQL Server Express, разработанная корпорацией Microsoft. Основной используемый язык запросов - Transact-SQL. Microsoft SQL Server Express нацелен в основном на встраиваемые и легковесные приложения;

- для визуализации онтологии использован специально разработанный программный интерпретатор на языке программирования C#, код которого приведен в Приложении, либо возможности Cytoscape.js - биоинформатической платформы с открытым исходным кодом.

Cytoscape.js – библиотека графиков jQuery, которая помогает визуализировать графики или сети, она оптимизирована, если дело касается преобразования сырых сетевых данных в графы и также может обрабатывать большие объемы данных.

Cytoscape.js совместима со всеми современными браузерами, CommonJS/NodeJS, jQuery и Meteor/Atmosphere.

На рисунке 5.1 представлены классы-контроллеры MVC образовательной среды.

Для работы с базой данных использована платформа Entity Framework - технология доступа к данным, являющаяся object-relational mapping (ORM) решением для .NET Framework от Microsoft. Данная технология предоставляет возможность взаимодействия с объектами посредством LINQ, в виде LINQ to Entities, так и с использованием Entity SQL.

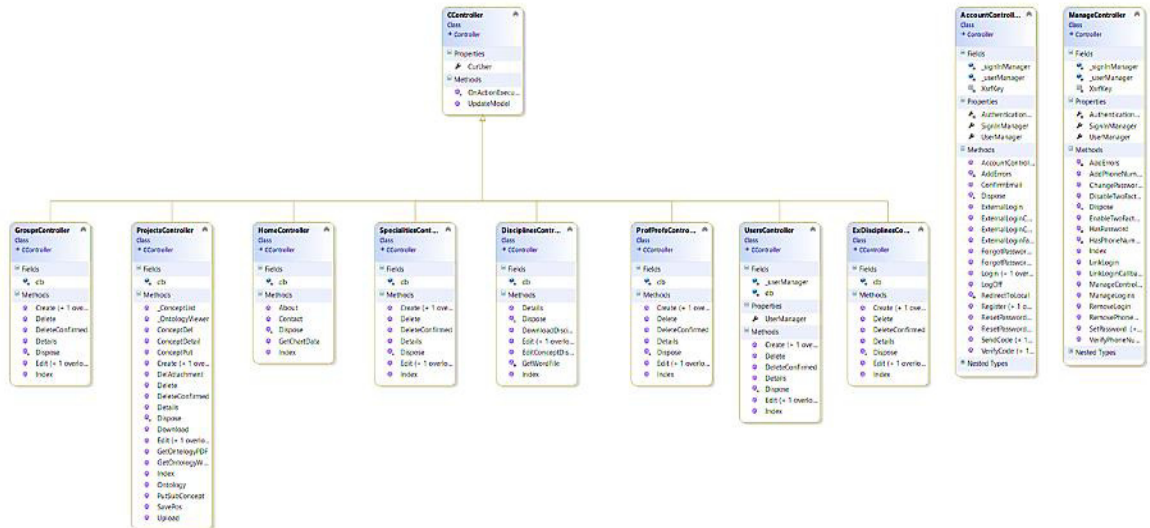


Рисунок 5.1 - Диаграмма классов-контроллеров MVC образовательной среды

На рисунке 5.2 представлена физическая модель базы данных образовательной среды.

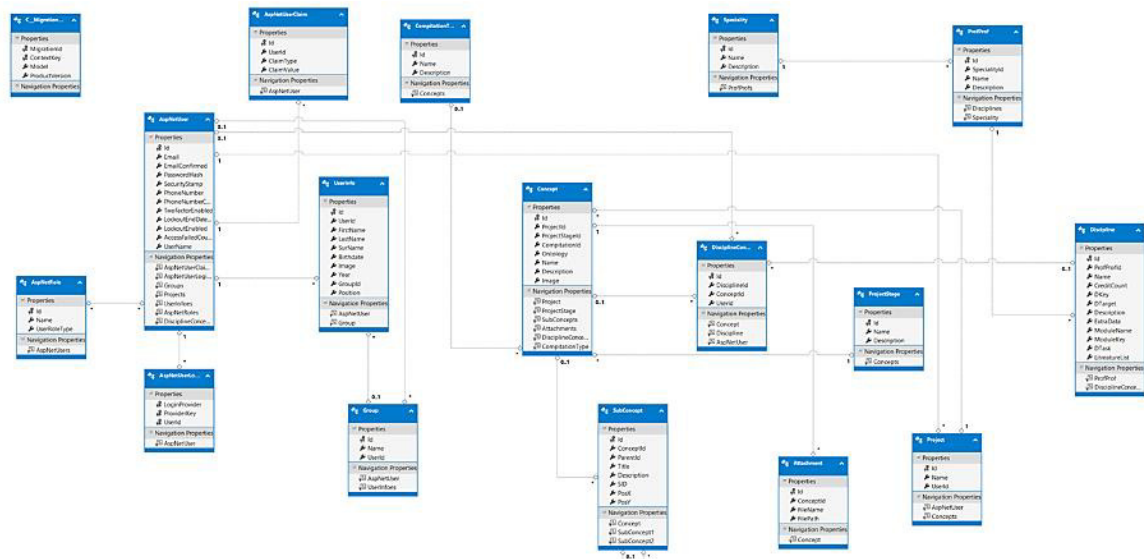


Рисунок 5.2- Инфологическая модель базы данных среды

Для создания динамических веб - страниц, с вставками серверного кода, использован Razor - движок представлений в ASP.NET.

Для проектирования веб - интерфейса среды использована jQuery-библиотека JavaScript, фокусирующаяся на взаимодействии JavaScript и HTML. Библиотека jQuery помогает легко получать доступ к любому элементу DOM, обращаться к атрибутам и содержимому элементов DOM и манипулировать ими. Библиотека jQuery также предоставляет удобный API для работы с AJAX.

В среде также использован Bootstrap - свободный набор инструментов для создания сайтов и веб - приложений, включающий в себя HTML - и CSS -

шаблоны оформления для типографики, веб - форм, кнопок, меток, блоков навигации и прочих компонентов веб - интерфейса, включая JavaScript - расширения.

Поскольку работа в студенческом проекте предполагает команду исполнителей, поэтому для управления совместной работы над проектом использован Team Foundation Server (TFS) - продукт корпорации Microsoft, представляющий собой комплексное решение, объединяющее в себе систему управления версиями, сбор данных, построение отчетов, отслеживание статусов и изменений по проекту.

Структуру компонентов среды, в нотации языка UML, иллюстрирует рисунок 5.3

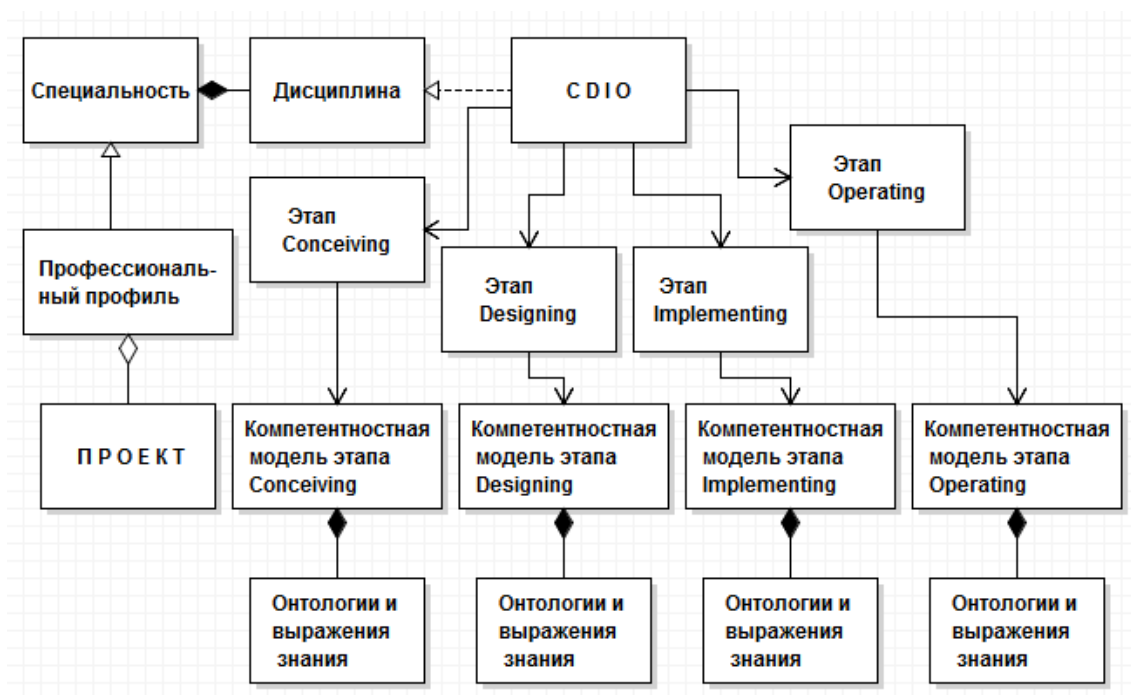


Рисунок 5.3 - Структура компонентов среды

Основными функциями эксперта являются:

1) Формирование списка проектов, составляющих контекст продуктивного проектно-организованного обучения и служащего для описания образовательных (знаниевых) компонентов дисциплин специальности.

2) Выбор этапа CDIO и организация бизнес-процессов и работ по разработке соответствующего проекта.

3) Описание дескриптора компетентностной модели - ДКМэ, и формирование концептуальной модели этапа - КМэ, включающей модель профессиональных компетенций - РК.КМэ; модель базовых компетенций - ВК.КМэ и модель дополнительных компетенций - ДК.КМэ. Каждая из моделей компетенций специфицируется наборами опорных понятий.

4) Для спецификации опорных понятий профессиональных, базовых и дополнительных компетенций используются выражения знаний, а для визуализации - ориентированный понятийный граф.

Выражение знания, являясь алгебраической моделью отражения знания, имеет удобный формат для хранения в базе данных среды и для операций над своими компонентами. Для визуализации онтологии опорного понятия можно использовать специально разработанный программный интерпретатор среды, либо возможности Cytoscape.js - биоинформатической платформы с открытым исходным кодом.

5) Формирование образовательных (знаниевых) компонент планируемого обучения CDIO Syllabus для профилирующих дисциплин KC.Syllabus.Prof и образовательных (знаниевых) компонент для базовых дисциплин - KC.Syllabus.Base учебного плана.

Сценарий работы эксперта представлен на рисунке Б-1. Итоговым результатом работы эксперта со средой будет заключаться в формировании двух файлов.

Первый файл - KC.Syllabus.Prof, содержит выражения знания профессиональных компетенций всех этапов инициативы CDIO, предназначенные для проектирования профилирующих дисциплин специальности.

Второй файл - KC.Syllabus.Base, содержит выражения знаний базовых и дополнительных компетенций всех этапов инициативы CDIO, предназначенные для проектирования базовых дисциплин специальности.

Онтологии и выражения знаний этих двух файлов предназначены для преподавателя, который должен будет использовать образовательные (знаниевые) компоненты при проектировании учебно-методического комплекса соответствующей дисциплины.

б) И, наконец, последняя функция эксперта - это проведение мониторинга внешних информационных ресурсов по направлению "Инженерия программного обеспечения" и другим смежным областям, с целью внесения необходимых изменений и дополнений в образовательные (знаниевые) компоненты базы данных среды.

Для работы пользователей со средой используется главная страница, представленная на рисунке 5.4, на которой показана статистика среды, отражающая количество активных проектов и дисциплин. На момент написания данного раздела в базе данных было сформировано 54 опорных понятия и по каждому из этапов CDIO показано количество опорных понятий для профессиональных, базовых и дополнительных компетентностей.

В левой части главной страницы расположены ссылки на компоненты среды для работы эксперта, преподавателя и администратора среды.

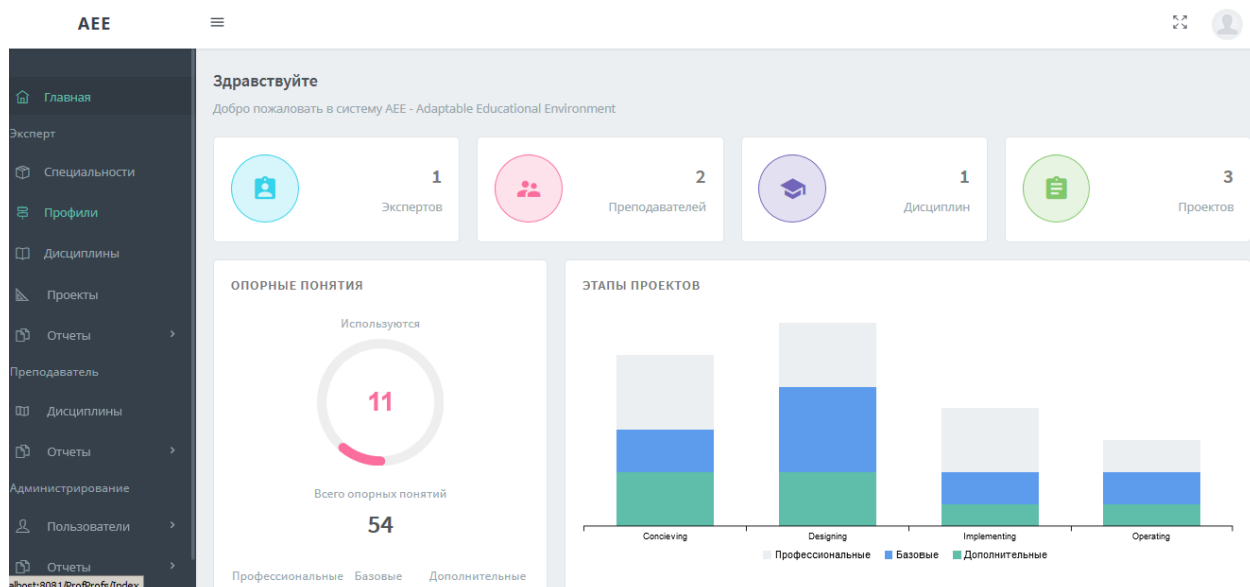


Рисунок 5.4 - Главная страница среды

Далее покажем сценарий работы эксперта со средой и соответствующие страницы представлений.

Первым делом эксперт должен выбрать один из профессиональных профилей, например, "Распределенные приложения и системы реального времени", специальности "Вычислительная техника и программное обеспечение" направления "Инженерия программного обеспечения", как показано на страницах, представленных на рисунках 5.5 и 5.6.

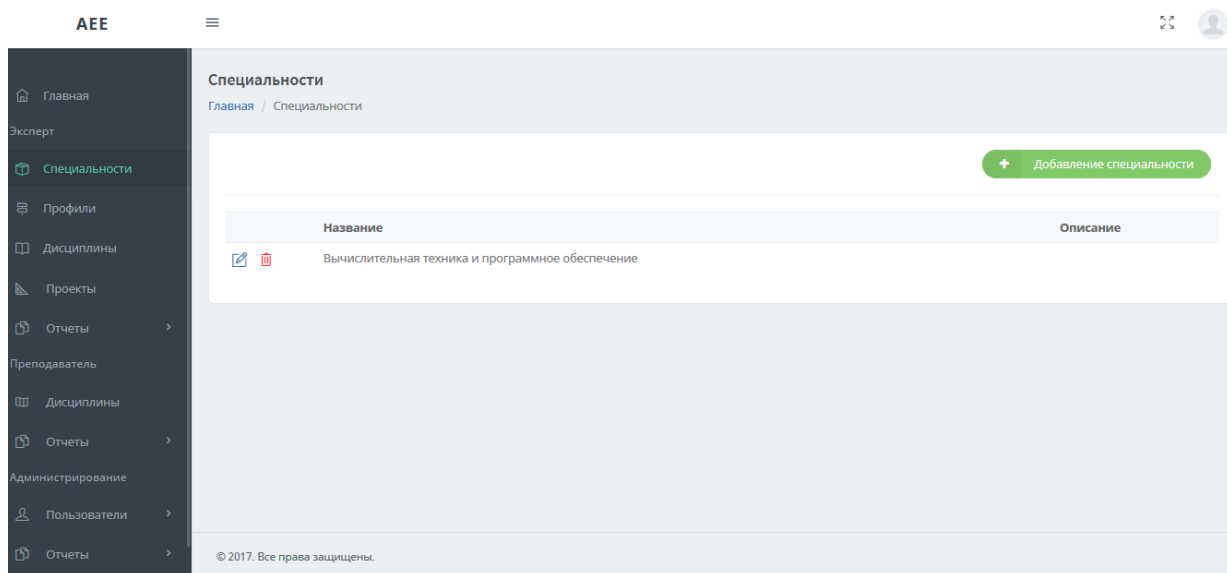


Рисунок 5.5- Страница работы эксперта - "Специальности"



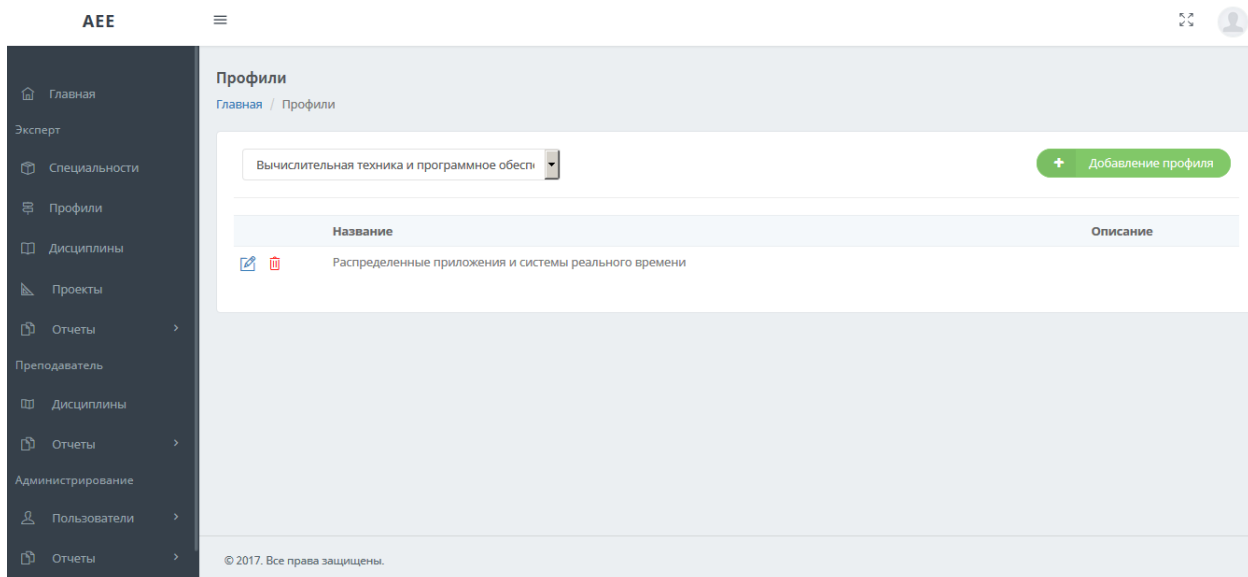


Рисунок 5.6 - Страница работы эксперта - "Профили"

Далее эксперт выбирает один из планируемых проектов (рисунок 5.7), составляющий контекст продуктивного проектно-организованного обучения, в рамках которого будут формироваться семантические знания проектируемой дисциплины учебного плана специальности.

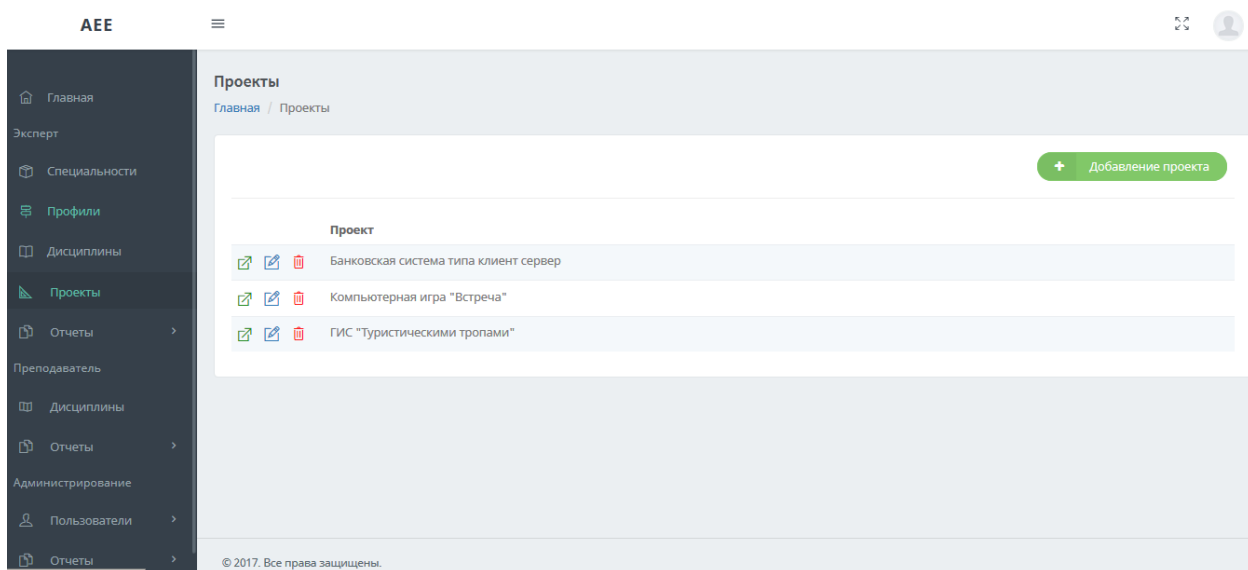


Рисунок 5.7 - Страница работы эксперта - "Проекты"

Следующие три страницы на рисунках 5.8, 5.9 и 5.10 показывают структуру и содержимое дескрипторов дисциплин. В данном примере речь идет о дескрипторе дисциплины "Технологии разработки распределенных приложений", для которой эксперт формирует образовательные (знаниевые) компоненты.

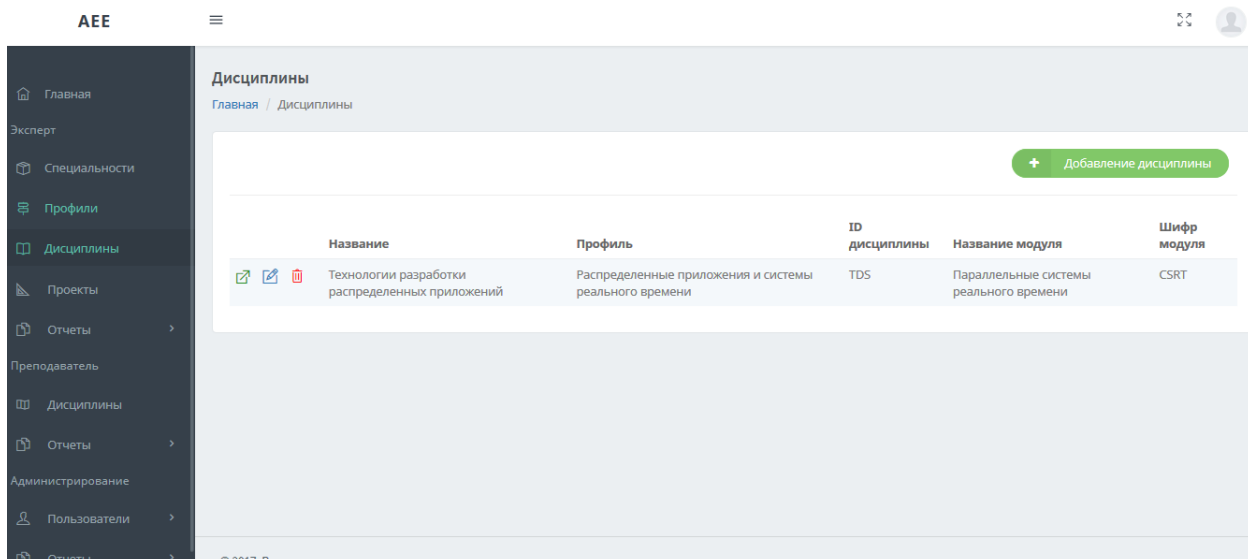


Рисунок 5.8 - Структура дескриптора дисциплины "Технологии разработки распределенных приложений"

На данной и последующих страницах показаны возможности среды по добавлению новой дисциплины, просмотру содержимого дескриптора, редактированию содержимого и удалению информации из содержимого текущей дисциплины, для чего в среде предусмотрены опции: "Просмотр", "Редактирование" и "Удаление информации".

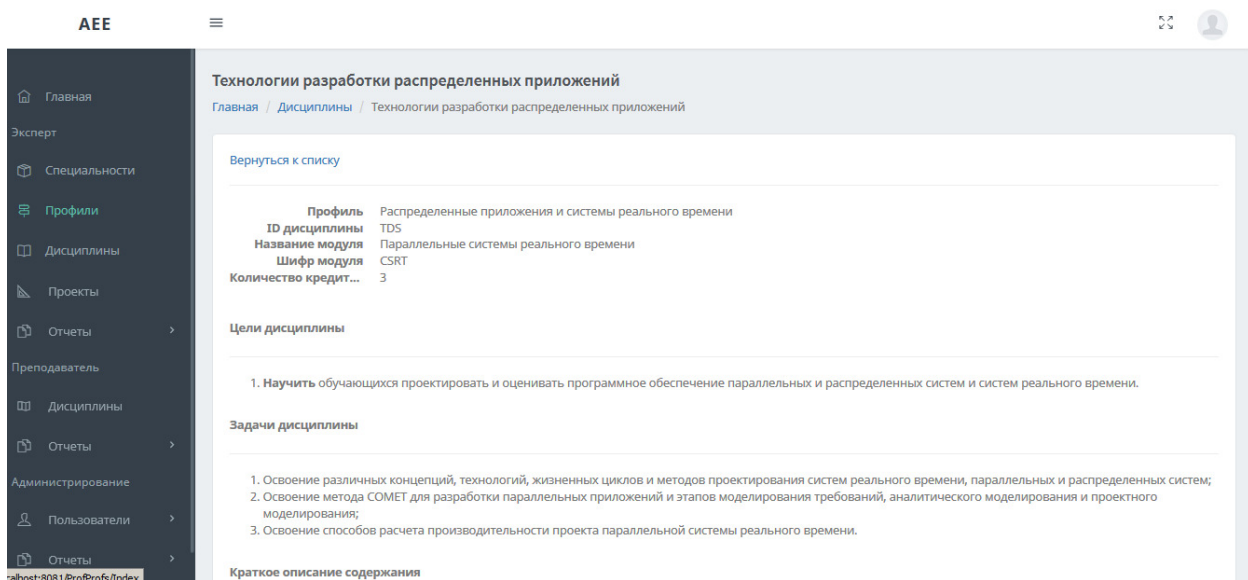


Рисунок 5.9 - Продолжение страницы - содержание дескриптора дисциплины "Технологии разработки распределенных приложений"

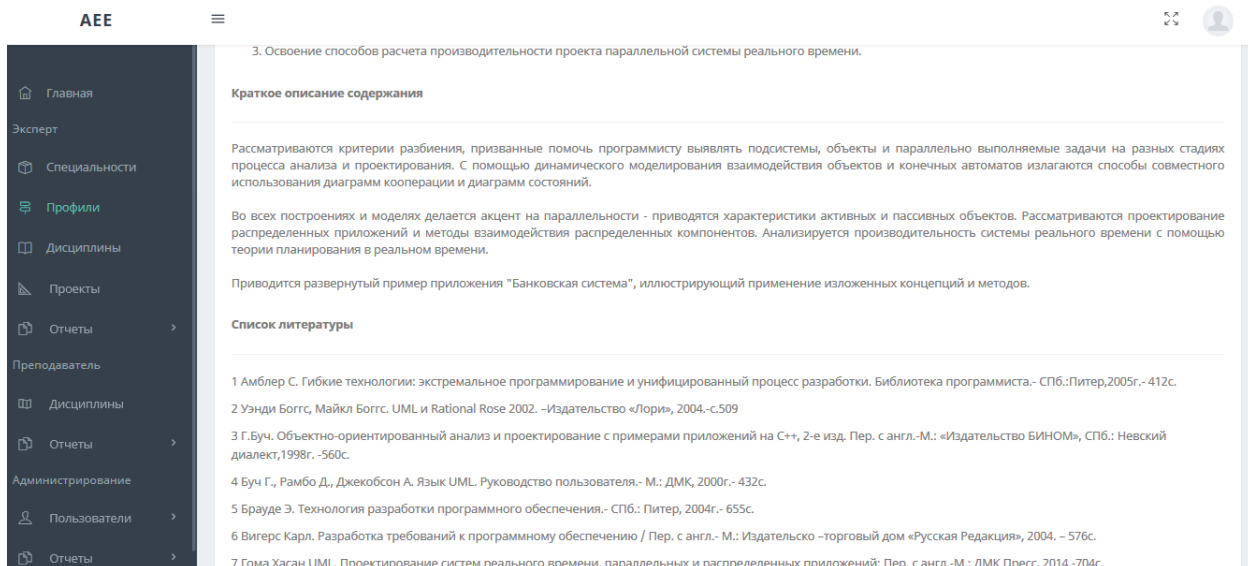


Рисунок 5.10 - Продолжение страницы - содержание дескриптора дисциплины "Технологии разработки распределенных приложений"

Далее для дисциплины "Технологии разработки распределенных приложений", в рамках проекта "Банковская система типа клиент-сервер", следует процесс разработки онтологий опорных понятий соответственно профессиональных, базовых и дополнительных компетенций каждого из этапов инициативы CDIO, с последующим вводом в базу данных среды, как проиллюстрировано на рисунках 5.11 и 5.12.

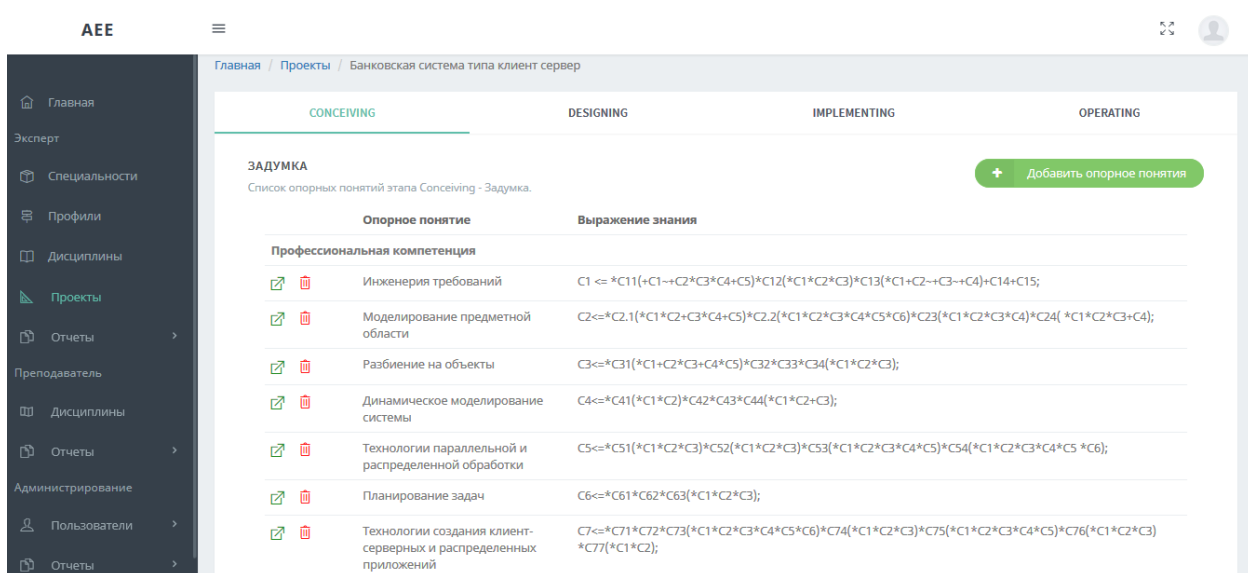


Рисунок 5.11 - Страница формирования и добавления опорных понятий профессиональных компетенций этапа Conceiving в базу данных среды

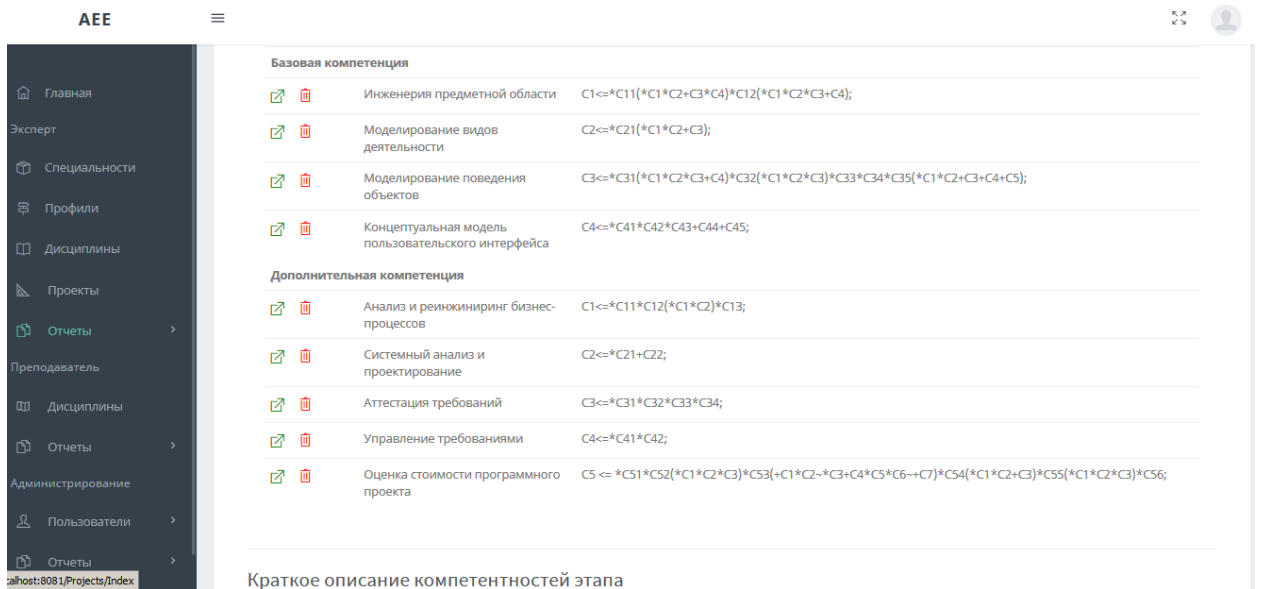


Рисунок 5.12 - Страница формирования и добавления опорных понятий базовых и дополнительных компетенций этапа Conceiving в базу данных среды

Далее выбирается этап Designing в рамках которого для каждой из компетентностных моделей последовательно вводятся названия опорных понятий и их выражения знания, как показано на рисунках 5.13, 5.14 и 5.15.

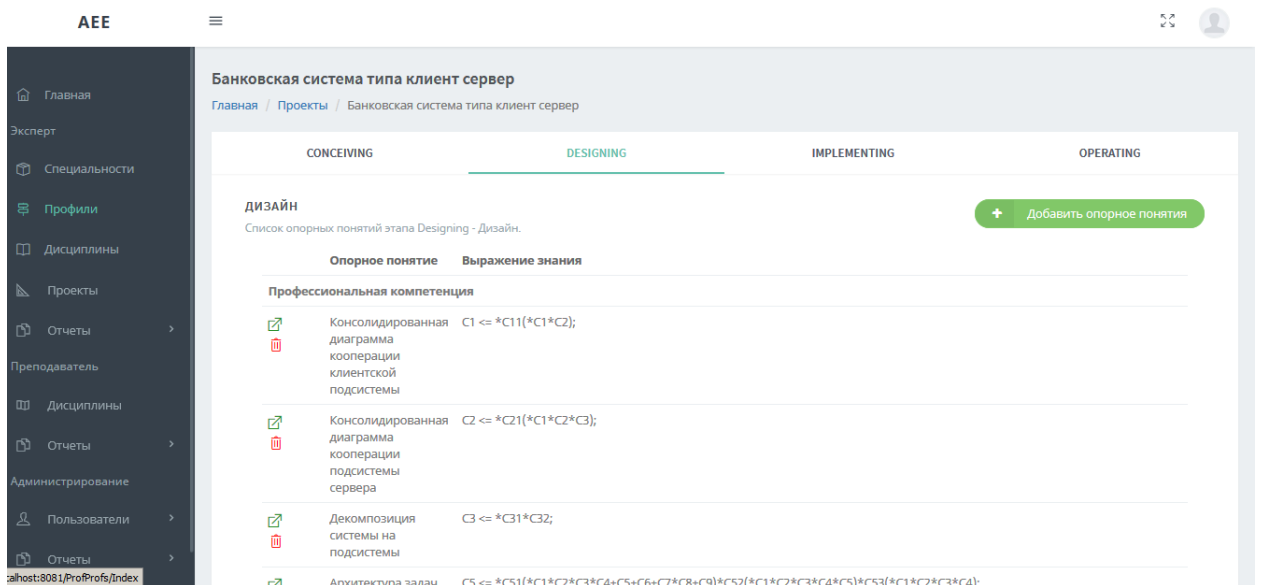


Рисунок 5.13 - Страница формирования и добавления опорных понятий профессиональных компетенций этапа Designing в базу данных среды

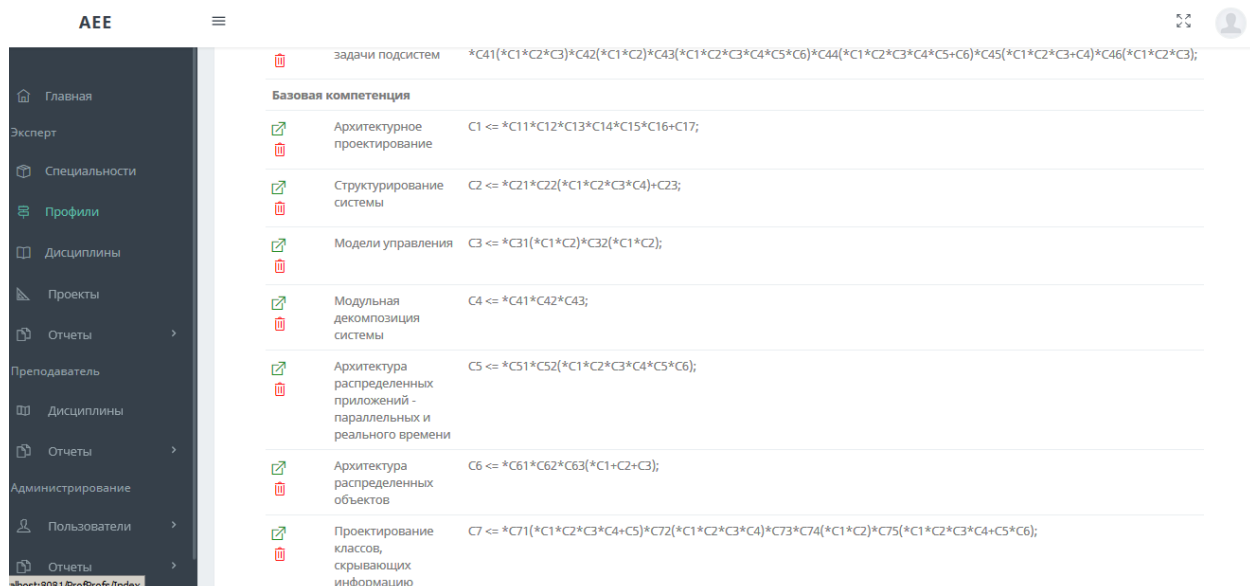


Рисунок 5.14 - Страница формирования и добавления опорных понятий базовых компетенций этапа Designing в базу данных среды

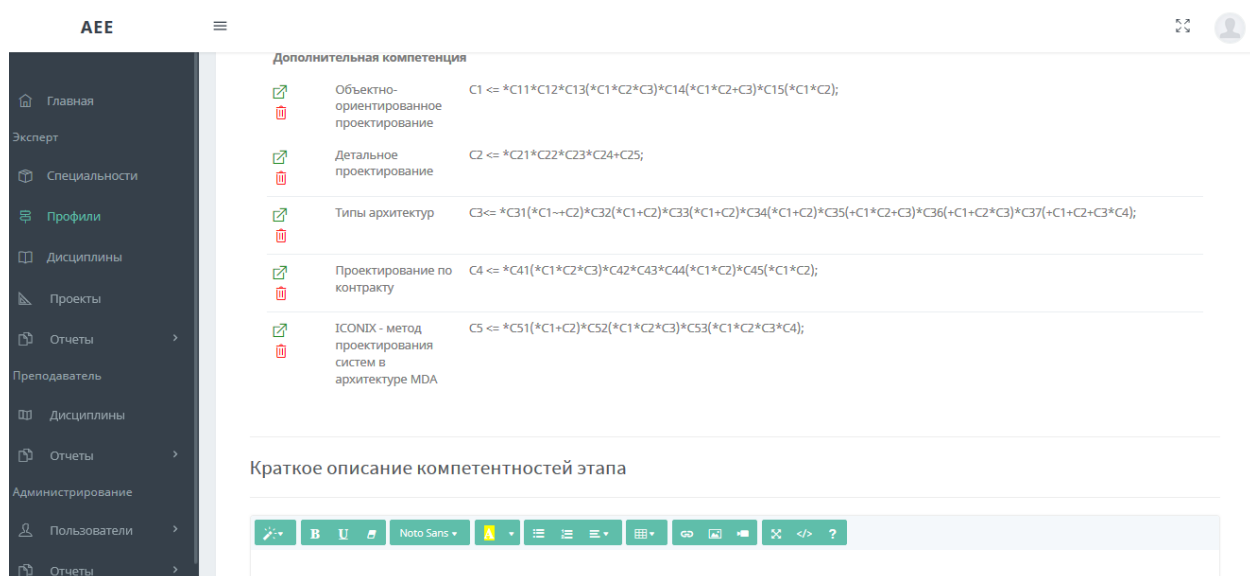


Рисунок 5.15 - Страница формирования и добавления опорных понятий дополнительных компетенций этапа Designing в базу данных среды

Таким же образом формируются опорные понятия и соответствующие выражения знаний двух оставшихся этапов инициативы CDIO, то есть этапов Implementing и Operating.

Для каждого из вводимых опорных понятий предусматривается их идентификация конкретизирующими понятиями первого и второго уровней, для чего используется опция "Редактирование" опорного понятия.

Рассмотрим работу среды на примере опорных понятий "Архитектура распределенных приложений - параллельных и реального времени" и

"Архитектура распределенных объектов", из набора базовых компетенций этапа Designing.

После выбора опции "Просмотр" указанных опорных понятий, среда покажет выражения знаний этих понятий и построит соответствующие понятийные графы каждого из них, как показано на рисунках 5.16, 5.17.

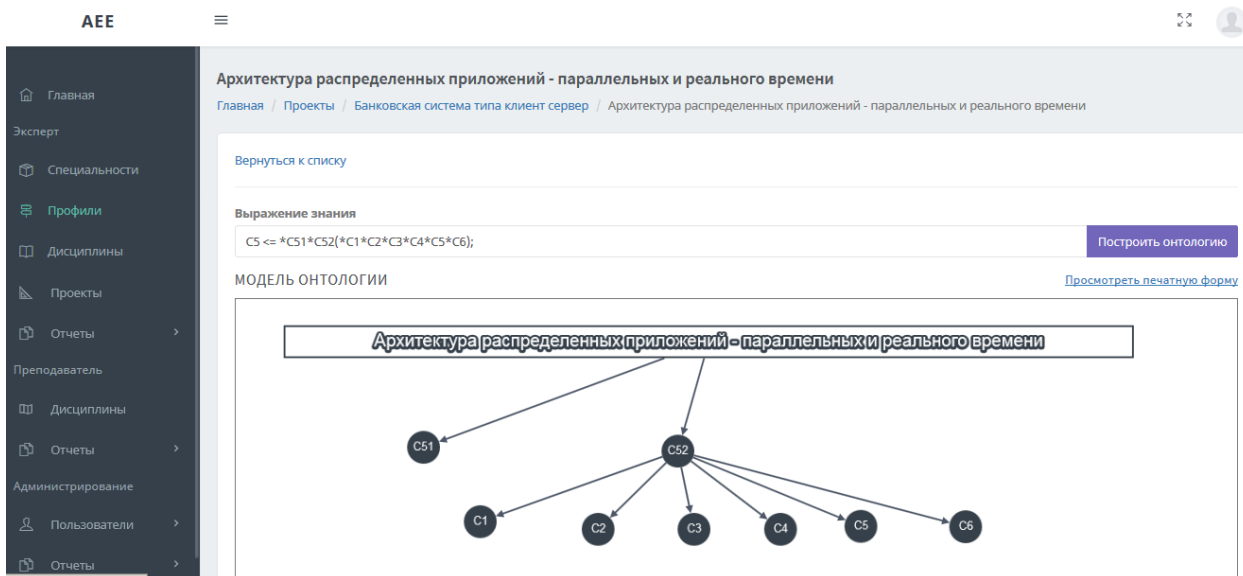


Рисунок 5.16 - Понятийный граф онтологии опорного понятия "Архитектура распределенных приложений - параллельных и реального времени"

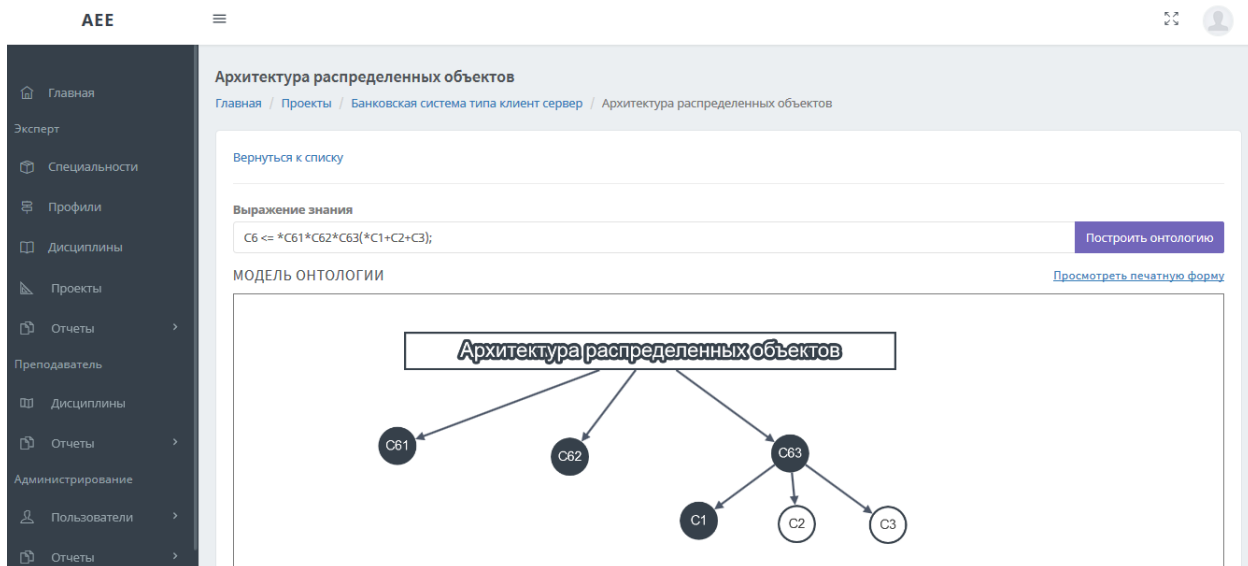


Рисунок 5.17 - Понятийный граф онтологии опорного понятия "Архитектура распределенных объектов"

Пусть, например, в онтологии опорного понятия "Архитектура распределенных объектов", выберем конкретизирующее понятие первого уровня -  $C_{63}$ , после клика на который среда покажет окно для ввода названия

этого конкретизирующего понятия и его краткого описания, как показано ниже на рисунке 5.18.

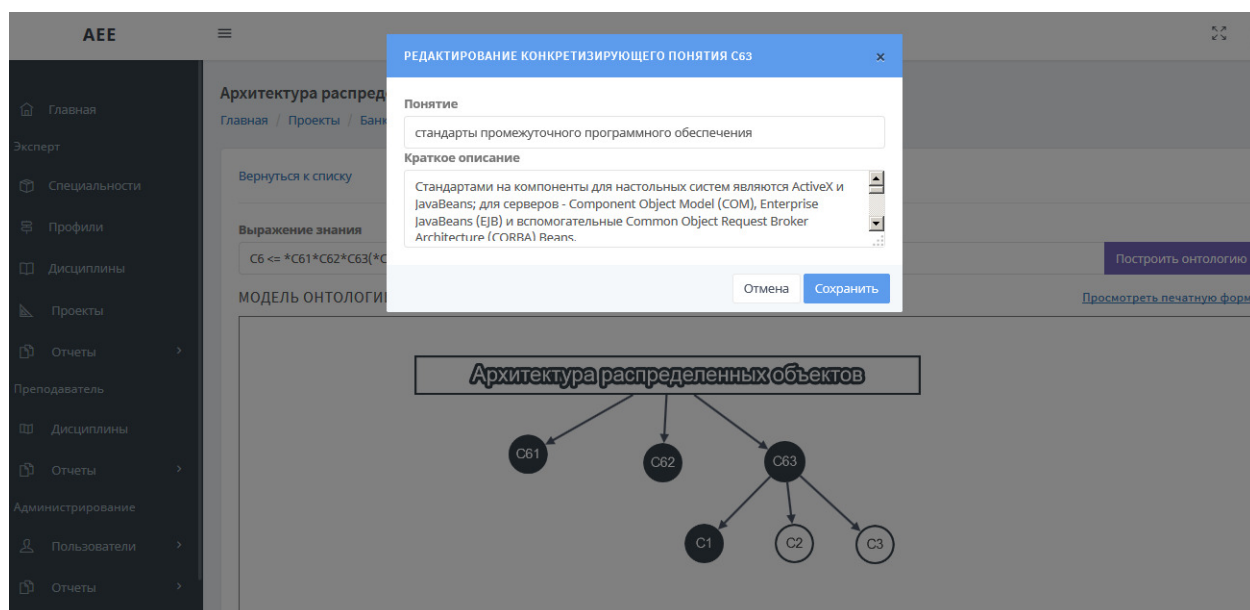


Рисунок 5.18 - Страница редактирования информации для конкретизирующего понятия С<sub>63</sub> - "Стандарты промежуточного программного обеспечения"

Таким образом происходит сеанс работы эксперта со средой по формированию базы данных из образовательных (знаниевых) компонент планируемого обучения CDIO Syllabus [62] для дисциплины "Технологии разработки распределенных приложений". Сформированные таким образом знаниевые компоненты должны будут использоваться преподавателем для проектирования учебно-методического комплекса этой дисциплины.

Роль проекта "Банковская система типа клиент-сервер" в рассмотренном примере, заключается в том, что он является продуктивным проектно-организованным контекстом, определяющим знаниевое содержимое компетентностных моделей дисциплины "Технологии разработки распределенных приложений", необходимое обучающимся при проведении бизнес-процессов и работ по указанному проекту.

## 5.2 Сценарий работы преподавателя

Работа преподавателя со средой сводится к использованию знаниевого содержимого компетентностных моделей для проектирования учебно-методического комплекса данной дисциплины [73].

Сеанс работы преподавателя начинается с выбора "Дисциплины", как показано на рисунке 5.19.

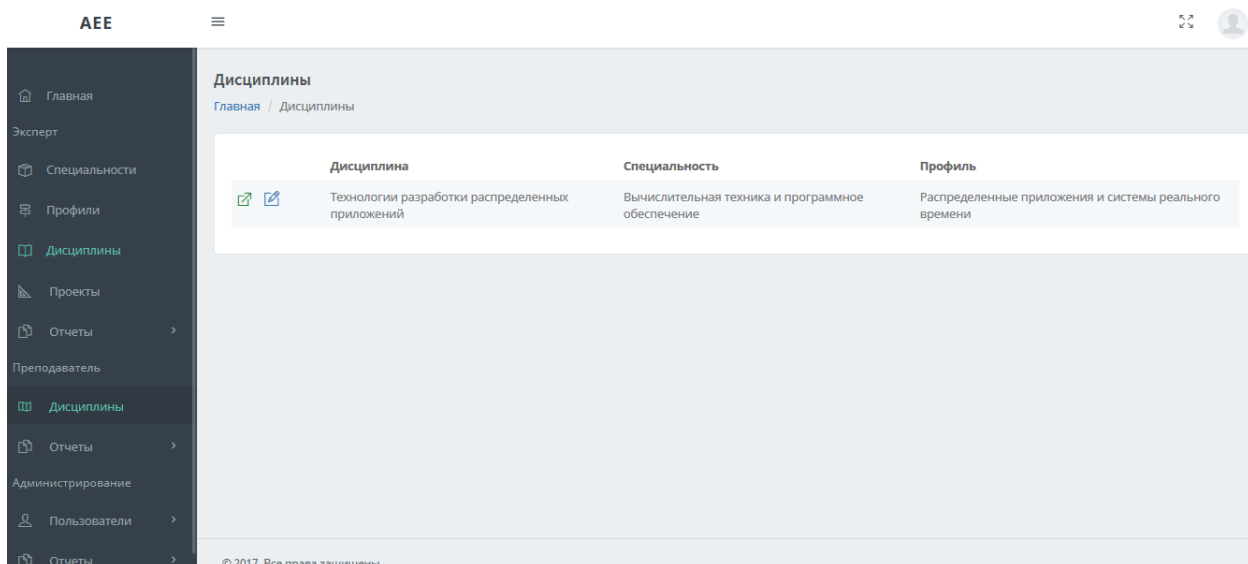


Рисунок 5.19 - Страница работы преподавателя со средой по выбору дисциплины

Далее, на страницах каждого из этапов и соответствующих компетенций преподаватель выбирает опорные понятия этапов, которые он намерен использовать при проектировании своей дисциплины, руководствуясь, в первую очередь компонентами профессиональных компетентностей, как показано на рисунках 5.20, 5.21, 5.22.

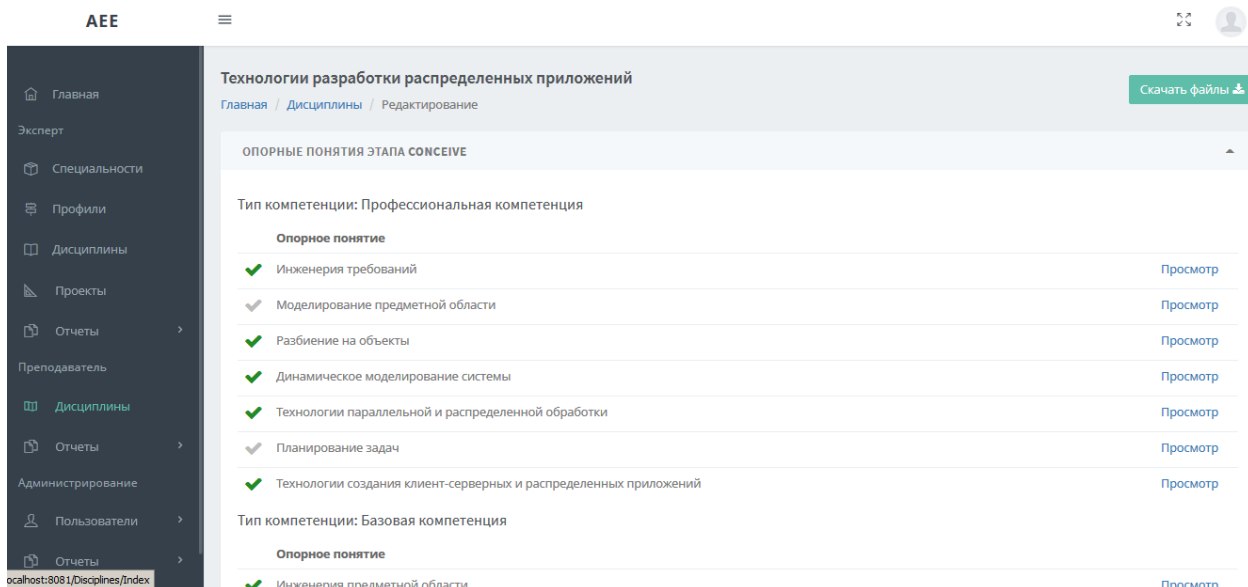


Рисунок 5.20 - Список опорных понятий профессиональных компетентностей этапа Conceive



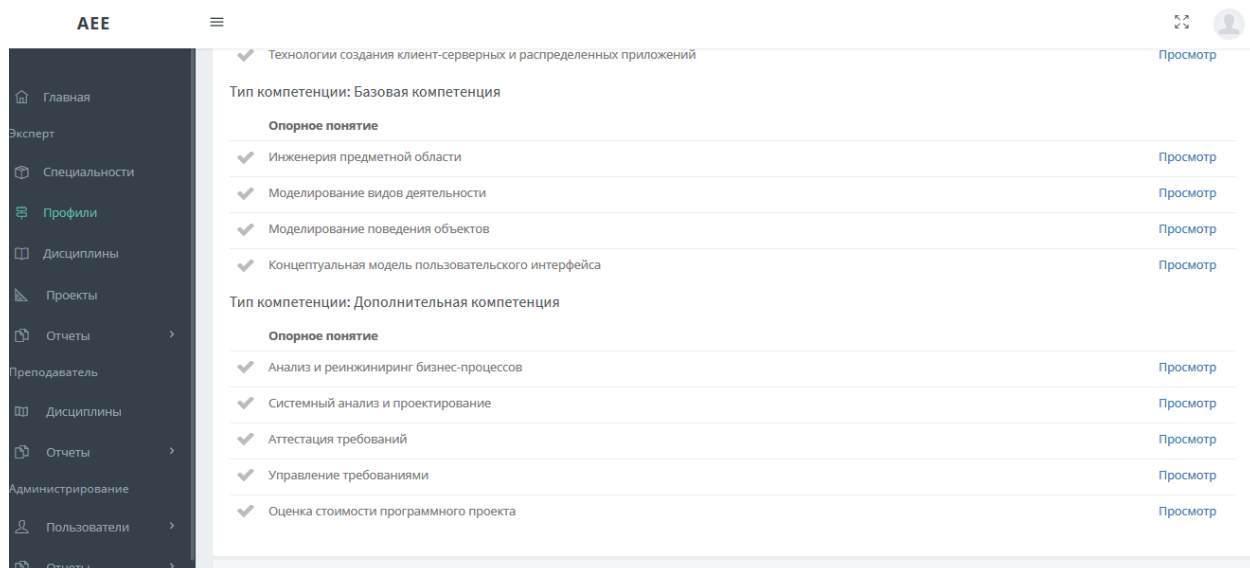


Рисунок 5.21- Список опорных понятий базовых и дополнительных компетентностей этапа Conceive

И так далее по всем этапам, после чего по опции "Скачать файлы" среда формирует текстовый файл с дескриптором дисциплины и печатными формами опорных понятий и два файла образовательных (знаниевых) компонент, состоящих соответственно из профессиональных компетентностей и из базовых и дополнительных компетентностей.

Ниже приводится содержимое сформированного текстового файла:

### **Технологии разработки распределенных приложений**

|                            |   |
|----------------------------|---|
| <b>Профиль</b>             | Распределенные приложения и системы реального времени |
| <b>ID дисциплины</b>       | TDS   |
| <b>Количество кредитов</b> | 3   |
| <b>Название модуля</b>     | Параллельные системы реального времени                |
| <b>Шифр модуля</b>         | CSRT  |

### **Цели дисциплины**

а) Научить обучающихся проектировать и оценивать программное обеспечение параллельных и распределенных систем и систем реального времени.

### **Задачи дисциплины**

- Освоение различных концепций, технологий, жизненных циклов и методов проектирования систем реального времени, параллельных и распределенных систем;
- Освоение метода СОМЕТ для разработки параллельных приложений и этапов моделирования требований, аналитического моделирования и проектного моделирования;
- Освоение способов расчета производительности проекта параллельной системы реального времени.

### 5.3 Экспериментальные результаты

Для проведения экспериментальной проверки эффективности моделей и методики представления и организации знаний с применением онтологического подхода приведен анализ основных сервисов [75], которые позволяют выполнять близкие по смыслу задачи (выявление именованных сущностей) обработки текстов на естественном языке в приложении к размеченным текстам на русском языке. Проанализированы возможности следующих систем: Rosette Text Analytics, Pullenti, TextRazor, Eureka Engine.

Цель эксперимента заключалась в том, чтобы оценить возможности существующих сервисов по выявлению ключевых или базовых понятий (концептов) в текстах, относящихся к анализируемой предметной области.

### 5.4 Метод исследования

Для проведения анализа использованы тесты лекций «Маршрутизация и коммутация», которые были аннотированы автором диссертации, размечены с указанием концептов. Лекции прошли контроль экспертов по структуризации исходного контента в виде кратких итогов, после чего выполнялась разметка текста по выделению концептов. Затем были сформированы онтологии с использованием разработанной для этих целей среды АЕЕ, что позволило согласовать концепты между собой и удостовериться в правильности разметки.

Пример размеченного текста, аннотирующего лекцию «Функции маршрутизаторов» и полученной на его основе онтологии приведен в приложении Б.

Тексты со снятой разметкой подавались на вход упомянутых систем.

Выполнен сравнительный анализ сервисов с использованием следующих показателей: доля правильных ответов сервиса, точность, полнота и гармоническое среднее (Accuracy, Precision, Recall и F1, соответственно), которые повсеместно применяются для оценки алгоритмов и программ классификации данных. Расчет указанных показателей базируется на следующих количественных параметрах:

- Количество правильно распознанных сущностей (tp - истинно положительные)
- Количество сущностей которые неверно выделены как сущности (fp - ложно-положительные),

– Количество реально имеющихся сущностей не распознанных сервисом (fn - ложно-отрицательные).

Отметим, что в задачах оценки алгоритмов классификации используется показатель tn равный количеству объектов противоположного класса правильно распознанных классификатором. Поскольку количество таких объектов неизвестно, то данный показатель во всех случаях принят равным 0. Исходя из принятых обозначений:

$$\text{Accuracy} = \frac{tp+tn}{tp+tn+fp+fn} \quad (5.1)$$

Показывает долю правильно распознанных сущностей к общему количеству сущностей. Ошибка первого рода:

$$\text{Precision} = \frac{tp}{tp+fp} \quad (5.2)$$

Показывает точность сервиса, то есть насколько часто он приписывает лишние сущности к имеющимся. Чем выше Precision сервиса, тем меньше лишних сущностей он определяет. Ошибка второго рода:

$$\text{Recall} = \frac{tp}{tp+fn} \quad (5.3)$$

Показывает насколько полно сервис определяет имеющиеся сущности. Чем выше Recall сервиса, тем меньше пропусков в распознавании именованных сущностей он допускает.

F мера (F1), представляет собой гармоническое среднее между точностью (precision - p) и полнотой (recall - r). Часто вычисляется с помощью следующего выражения:

$$F_1 = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} \quad (5.4)$$

Сравнение качественных характеристик сервисов выполнено на базе 8 текстов лекций, описание которых приведено по адресу <https://www.intuit.ru/studies/courses/3646/888/lecture/31151>. Для эксперимента использовался курс лекций, посвящённый изложению принципов маршрутизации и коммутации в сетях пакетной коммутации. В курсе рассматриваются примеры конфигурирования протоколов маршрутизации, виртуальных локальных сетей, списков контроля доступа, трансляторов адресов, и другие технологии.

## 5.5 Результаты сравнительного анализа облачных сервисов по выявлению концептов

Таблица 5.1 Анализ характеристик сервисов

| Количественные параметры          | Показатели | Эксперты | textrazor | Rosette Text Analytics | Eureka Engine | Pullenti |
|-----------------------------------|------------|----------|-----------|------------------------|---------------|----------|
| 1 лекция                          |            |          |           |                        |               |          |
| Правильно распознанные сущности   | tp         | 43       | 16        | 4                      | 10            | 43       |
| Не верно выделенные               | fp         | 0        | 28        | 0                      | 3             | 3312     |
| Реально имеющиеся не распознанные | fn         | 0        | 27        | 39                     | 33            | 0        |
| 2 лекция                          |            |          |           |                        |               |          |
| Правильно распознанные сущности   | tp         | 16       | 8         | 0                      | 3             | 16       |
| Не верно выделенные               | fp         | 0        | 15        | 0                      | 3             | 263      |
| Реально имеющиеся не распознанные | fn         | 0        | 8         | 16                     | 16            | 0        |
| 3 лекция                          |            |          |           |                        |               |          |
| Правильно распознанные сущности   | tp         | 33       | 10        | 4                      | 9             | 33       |
| Не верно выделенные               | fp         | 0        | 25        | 5                      | 5             | 442      |
| Реально имеющиеся не распознанные | fn         | 0        | 23        | 29                     | 24            | 0        |
| 4 лекция                          |            |          |           |                        |               |          |
| Правильно распознанные сущности   | tp         | 28       | 14        | 7                      | 8             | 28       |
| Не верно выделенные               | fp         | 0        | 13        | 6                      | 12            | 275      |
| Реально имеющиеся не распознанные | fn         | 0        | 14        | 21                     | 20            | 0        |
| 5 лекция                          |            |          |           |                        |               |          |
| Правильно распознанные сущности   | tp         | 16       | 6         | 1                      | 2             | 16       |
| Не верно выделенные               | fp         | 0        | 7         | 1                      | 0             | 226      |
| Реально имеющиеся не распознанные | fn         | 0        | 10        | 15                     | 14            | 0        |
| 6 лекция                          |            |          |           |                        |               |          |
| Правильно распознанные сущности   | tp         | 36       | 10        | 6                      | 5             | 36       |
| Не верно выделенные               | fp         | 0        | 27        | 4                      | 1             | 426      |
| Реально имеющиеся не распознанные | fn         | 0        | 26        | 30                     | 31            | 0        |
| 7 лекция                          |            |          |           |                        |               |          |
| Правильно распознанные сущности   | tp         | 15       | 8         | 2                      | 4             | 15       |
| Не верно выделенные               | fp         | 0        | 27        | 0                      | 9             | 288      |
| Реально имеющиеся не распознанные | fn         | 0        | 7         | 13                     | 11            | 0        |
| 8 лекция                          |            |          |           |                        |               |          |
| Правильно распознанные сущности   | tp         | 21       | 10        | 3                      | 3             | 21       |
| Не верно выделенные               | fp         | 0        | 27        | 13                     | 12            | 234      |
| Реально имеющиеся не распознанные | fn         | 0        | 11        | 18                     | 18            | 0        |

Таблица 5.2 Результаты эксперимента

| Средние результаты | textrazor | Rosette Text Analytics | Eureka Engine | Pullenti |
|--------------------|-----------|------------------------|---------------|----------|
| A                  | 0,23      | 0,10                   | 0,16          | 0,06     |
| P                  | 0,34      | 0,53                   | 0,58          | 0,06     |
| R                  | 0,42      | 0,12                   | 0,20          | 1,00     |
| F                  | 0,37      | 0,18                   | 0,28          | 0,12     |

В результате анализа выявлены лидеры по отдельным показателям:

- по доли правильных ответов алгоритма лидером является Text Razor,
- по точности сервисов лидером является Eureka Engine,
- по полноте сервисов - Pullenti,
- по гармоническому среднему- TextRazor.

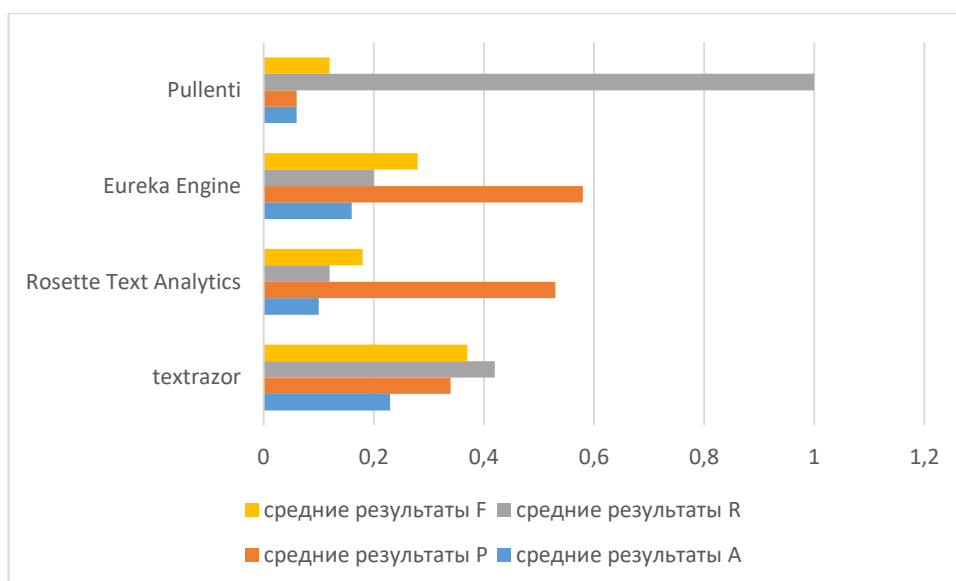


Рисунок 5.22 Результаты анализа

Полученные результаты, представленные на рисунке 5.22, говорят о том, что рассматриваемые облачные сервисы не могут обеспечить необходимое качество выявления концептов знаний требуемых для формирования онтологий. Что делает актуальным разработку программы автоматизированного формирования онтологий, представленной в данной диссертации.

Вместе с тем, несмотря на низкие текущие результаты, можно предположить, что рассматриваемые сервисы, в случае использования специальной библиотеки сущностей из заданной предметной области, имели бы лучшие оценки качества [76].

Такие библиотеки сущностей в форме концептов также могут быть подготовлены с использованием разработанной программы

Предложенный алгоритм и программная реализация, разработанная в рамках исследования, по существу обеспечивают работу по подготовке размеченного корпуса сущностей для возможных сервисов.

### **Выводы по 5 разделу**

Разработанный язык спецификации знаний наделен достаточными возможностями по анализу и описанию структуры информации и релевантному отображению семантики образовательных ресурсов; язык позволяет учитывать свойства общности и изменчивости понятий, что дает возможность конфигурировать знаниевые компоненты многократного применения, с их дальнейшим использованием для конструирования учебных дисциплин и индивидуальных образовательных программ.

Рассмотрены критерии разбиения, призванные помочь программисту выявлять подсистемы, объекты и параллельно выполняемые задачи на разных стадиях процесса анализа и проектирования программных систем. С помощью динамического моделирования взаимодействия объектов и конечных автоматов изложены способы совместного использования проектных диаграмм.

Результаты проведенного эксперимента, следующих систем: Rosette Text Analytics, Pullenti, TextRazor, Eureka Engine показали, что данные облачные сервисы не могут обеспечить необходимое качество выявления концептов знаний, требуемых для формирования онтологий, что делает актуальным разработку программы автоматизированного формирования онтологий, представленной в данной диссертации.

Во всех построениях и моделях сделан акцент на параллельности - приведены характеристики активных и пассивных объектов и анализ производительности системы реального времени с помощью теории планирования в реальном времени.

## ЗАКЛЮЧЕНИЕ

На сегодняшний день разработано уже достаточное количество моделей отображения знаний. Каждая из них обладает своими плюсами и минусами, и поэтому для каждой конкретной задачи необходимо выбрать именно свою модель. От этого будет зависеть не столько эффективность выполнения поставленной задачи, сколько возможность ее решения вообще.

Для реализации целей исследования решены следующие задачи:

1) Исследованы и разработаны эмпирические модели формализации знаний предметной области, моделей спецификации знаний на основе онтологического подхода, технологий и инструментальных средств.

2) Исследованы и разработаны модели формальных систем и процедур вывода знаний образовательной парадигмы, развиваемой в рамках компетентностной модели и метода проектного обучения Всемирной инициативы CDIO.

3) Проведены анализ и адаптация моделей знаниевых компонент информационно-образовательной среды.

4) Проведен анализ и обоснование проектных решений по аппаратно-программному обеспечению информационно-образовательной среды.

5) Проведена опытно-экспериментальная проверка эффективности работы информационно-образовательной среды.

6) Разработан образовательный сайт для популяризации информационно-образовательной среды, дальнейшего развития и совершенствования методики компетентностного подхода, технологий и инструментальных средств.

В диссертации рассмотрены основные подходы к построению моделей представления знаний, в каждом из которых есть свои достоинства и свои недостатки, которые делают каждую из них наиболее эффективной в конкретной области и при определенных условиях. Однако, после рассмотрения этих моделей становится очевидно, что создание интеллекта, подобного человеческому в плане самостоятельных обработки информации и использования знаний, пока невозможно. По крайней мере, не в чистых моделях. Логично предположить, что требуется совместить некоторые из этих моделей для получения основных качеств, свойственных человеческому разуму. Но даже в случае создания подобного интеллекта останется еще несколько качеств, которые практически невозможно запрограммировать - такие как неординарность мышления и способность к творчеству, умение строить предположения и догадки, создавать теории и идеи. И, как ни странно, именно эти качества в свое время подтолкнули человечество к идее создания искусственного интеллекта.

Одним из направлений исследований, имеющих целью описание функционирования сложных систем, способов принятия решений и управлений такими системами является создание интеллектуальных систем, основанных на знаниях, в частности – экспертных систем, подменяющих

человека-эксперта.

Выбранная в исследовании концепция представления знаний, основанная на сетевой модели, и разработанный язык спецификации знаний, позволяют формализовать семантический контекст образовательных ресурсов в виде повторно используемых знаниевых компонентов.

Изложенная методика и технология разработки знаниевых компонент, основана на концепциях проектно-компетентностной модели и этапах CDIO. Разработка образовательных систем, основанных на предлагаемой парадигме представления знаний, является в высшей степени трудоемким делом, а методика, облегчающая их разработку, может оказаться весьма востребованной для перехода на инновационный формат организации образовательных программ и процессов.

Таким образом, в диссертационной работе представлено решение проблемы перехода на новую информационную базу обучения, основанную на проектно-компетентностной парадигме организации знаний и онтологическом инжиниринге, с точки зрения системного видения трехуровневого высшего инженерного образования. Переход от традиционной информационной базы обучения к инженерии знаний всех уровней системы инженерного образования, предполагает внедрение инновационных моделей отображения знаний, в основу проектирования которых должен принят проектный метод обучения, согласно инициативы CDIO, и представление каждого из этапов инициативы соответствующей компетентностной моделью. Декларативные знания, как основа компетентностной модели каждого из этапов CDIO, представляются необходимым и достаточным набором онтологий опорных понятий семантического контекста образовательных ресурсов.

Разработанный язык спецификации знаний наделен достаточными возможностями по анализу и описанию структуры информации и релевантному отображению семантического контекста образовательных ресурсов. Кроме того, язык позволяет учитывать свойства общности и изменчивости понятий, что дает возможность конфигурировать знаниевые компоненты многократного применения, с их дальнейшим использованием для конструирования учебных дисциплин и индивидуальных образовательных программ.

В диссертационной работе проведен критический анализ современных образовательных и информационных технологий, теорий и методик повышения качества обучения. В результате проведенного анализа:

- выявлена общая тенденция необходимости перехода на новую парадигму информационной базы обучения, так как существующая база не предоставляет в должной мере возможности по диверсификации образовательных технологий и услуг;

- выявлено мнение исследователей ближнего и дальнего зарубежья о необходимости создания интеллектуальных программных систем в образовании, и переходу на Smart-обучение, а также в необходимости



интеллектуализации представления учебного контента;

- выявлено, что относительно учебно-методического обеспечения, явно прослеживается тенденция перехода на проектный метод обучения и формирование компетентностной модели выпускника;

- выявлено, что достаточно остро ставится проблема планирования, мониторинга и управления процессами проектирования учебных планов и программ, а также необходимость индивидуализации траекторий обучения, и применения Smart-технологий;

- проведено обоснование и выбор формализмов семантического структурирования понятий предметной области, на основе онтологии, и способов построения абстракций предметной области с использованием свойств общности и изменчивости;

- введены формализмы отображения знаний, на основе сетевой модели, в качестве реализации которой, принята модель онтологии;

- представлена методика формирования знаниевых компонент репозитория образовательной среды, на базе концепций проектного метода обучения, этапов инициативы CDIO, компетентностной модели выпускника и онтологического инжиниринга.

В качестве платформ для разработки знаниевых компонент и для проектирования дисциплин и учебных планов обучения, проведено обоснование по программно-аппаратной инфраструктуре сервера АЕЕ; разработан язык спецификации онтологии и представления онтологии в виде выражения знания, разработан парсер для редактирования и сборки онтологий. Для визуализации онтологий предусмотрено также использование возможностей Cytoscape.js - биометрической платформы с открытым исходным кодом.

В целом, следует отметить, что полученные в диссертации результаты имеют научную новизну и значимость, поскольку охватывают основанную на знаниях парадигму знаний в целом. В диссертации затрагиваются новые области научных проблем, которые должны применяться конкретно для образования Казахстана. Результаты исследований могут оказать существенное влияние как на экономические, так и на социальные аспекты, такие как возможное массовое внедрение интеллектуальных информационно-образовательных условий для нового (сетевое) поколения, а также качественное изменение содержания образования, методов и инструментов. Кроме того, результаты диссертации будут способствовать развитию гибкого обучения в интерактивной образовательной среде, персонализации и адаптации обучения, свободного доступа к образовательному контенту.

Таким образом, разработанная методика является стратегически важной в области науки и инженерного образования, что, вероятно, спровоцирует инновационное влияние на исследовательское сообщество в направлении применения интеллектуальных информационных систем.

Диссертационная работа фокусируется на разработке и развитии

моделей представления и организации знаний с использованием компетентностно-ориентированного и онтологического подходов, что приведет к разработке методологии по разработке курсов и учебных планов на основе проектного метода обучения и концепций всемирной инициативы CDIO, что представляет собой интересную и сложную проблему, очень важную в области науки и инженерного образования.

В заключении необходимо отметить, что диссертационная работа и результаты исследований имеют достаточно платформу для коммерциализации продукта. В диссертации выдвинуты новые направления в области инженерного образования, которые планируется выполнить в рамках гранта Комитета Науки Министерства Образования и Науки Республики Казахстан на 2018-2020г, по проекту № 0118PK00187 «Исследование и разработка моделей и методики представления и организации знаний с применением онтологического подхода и инструментальных средств Smart-технологии, при реализации образовательных программ и процессов». Руководитель к.т.н., доцент РГП на ПХВ "Институт информационных и вычислительных технологий" КН МОН РК, Кубеков Б.С.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Дегтярев С.Н. Онтологический подход к развитию креативности как основа когнитивной образовательной технологии // Вестник Тюменского государственного университета. - 2013. - №29. - С.114-122.

2 Топоркова О.М. Семиотико-онтологическая модель обучения как концептуальная основа организации учебного процесса // Прикладная информатика. - 2009. - № 4 (22). - С. 100-113.

3 Черняховская Л.Р., Давлетбаева А.Р., Хабибуллина Л. Н. Онтологический анализ требований к качеству электронного обучения, // Информационные технологии в образовании. - 2014. - № 22. -С. 143-149.

4 Шполянская И.Ю., Мирошниченко И.И. Онтологическая модель информационно-образовательной среды учебного подразделения вуза // Вестник Ростовского государственного экономического университета (РИНХ). - 2010. - № 31. - С. 201-209.

5 Титенко С.В. Онтологически-ориентированная система управления контентом информационно-учебных web-порталов // Образование и общество. - 2012. - Т.15, № 3. - С. 522-533.

6 Ермаков А.Е. Автоматизация онтологического инжиниринга в системах извлечения знаний из текста // Труды Междунар. конф. Диалог'2008. – М.: Наука, 2008. - С. 136-140.

7 Цуканова Н.И. Онтологическая модель представления и организации знаний: учебное пособие для вузов. - М.: Горячая линия-Телеком, 2015.- 272 с.

8 Гаврилова Т.А. Онтологический подход к управлению знаниями при разработке корпоративных систем автоматизации// Новости искусственного интеллекта.-2003.-№2.- С.24-30.

9 Бениаминов Е.М., Болдина Д.М. Система представления знаний Ontolingua - принципы и перспективы // НТИ Сер. 2, Информ. процессы и системы. 1999. №10.- С.14-28.

10 Мутанов, Г.М., Шевчук Е.В. Экспертная система оценки знаний методом тестирования: научное издание / - Омск : 2001. - 151с.

11 Ахметов Б.С. Технологии ведущих IT-вендоров в учебном процессе КазНТУ имени К.И.Сатпаева Труды Международной научно-практической конференции «Информационные и телекоммуникационные технологии: образование, наука, практика». - Алматы: Изд-во КазНТУ им. К.И. Сатпаева.- 2012.-С.102-112

12 Саданова, Б.М., Султанова, Б.К. Представление корпоративных знаний в ИС: Для студ. спец. 050704 - Вычислительная техника и программное обеспечение. - Караганда: КарГТУ, 2008. - 14с.

13 Шарипбай А. А., Кабенов Д. И., Разахова Б. Ш., Зулхажав А., Онтологическое моделирование знаний для решения задач предметной области // Всероссийская конференция с международным участием "Знания - Онтологии - Теории" (ЗОНТ-2015), Новосибирск, 6-8 октября 2015г.. -

Новосибирск, Институт математики им. С.Л. Соболева СО РАН. - 2015. - Том 1, - С. 111- 117.

14 Кубеков Б.С. Технологии разработки программного обеспечения: учебник. - Алматы: Экономика, 2011.- 307с.

15 Кубеков Б.С., Утепбергенов И.Т. Системные основания компетентностной модели IT-специалиста. //Матер. Десятой открытой Всерос. конф. Сб. «Преподавание информационных технологий в Российской Федерации». – М.: МГУ им.М.В. Ломоносова, 2012. - С.79-80.

16 Кубеков Б.С. Визуальные понятийные модели в обучении IT-специалиста. //Матер. Десятой открытой Всерос. конф. Сб. «Преподавание информационных технологий в Российской Федерации». –М.: МГУ им. М.В.Ломоносова, 2012.- С. 277-279.

17 Кубеков Б.С. Особенности начального обучения программированию. // Матер. Междунар. науч.-практ. конф. "Информационные и телекоммуникационные технологии: образование, наука, практика". -Алматы.: Института информационных и телекоммуникационных технологий КазНТУ, 2012. - С.136-138.

18 Кубеков Б.С. Инновационная концепция обучения программированию //Вестник университета «Туран».-2013.- №1 (57).- С.144-148.

19 Кубеков Б.С., Утегенова А.У. Онтологический инжиниринг в представлении и организации учебного контента // Вестник университета «Туран».-2015.- №1 (65).- С.264-269.

20 Кубеков Б.С., Утегенова А.У., Зыкин С.Л. Концепция парадигмы в задаче представления и организации знаний. //Матер. Междунар. науч.-практ. конф. "Информатизация образования: тенденции, перспективы, инновации" (ИТО-КФО 2015).- Алушта, Республика Крым, 2015.- С.25-30.

21 Kubekov B., Utepbergenov I. The use of multiparadigm approach to knowledge modeling.- // 7th International Conference on Education and New Learning Technologies".- Barselona (Spain), 2015.- P. 136-141.

22 Kubekov B., Kuandykova J., Utepbergenov I., Utegenova A. Application of the conceptual model of knowledge for formalization of concepts of educational content //9th International Conference on Application of Information and Communication Technologies AICT2015.- Rostov-on-Don, 2015.- P.294-306.

23 Kubekov B., Utegenova A., Naumenko V. Applying of ontological engineering to represent knowledge and training sessions // 10th International Conference on Application of Information and Communication Technologies - AICT2016. – Baku, Azerbaijan, 2016. - P.115-118.

24 Kubekov B. Educational components formation technology for the planned CDIO SYLLABUS education // 9th annual International Conference of Education, Research and Innovation - ICERI2016. – Seville, Spain, 2016. - P.6139-6145.

25 Кубеков Б.С., Утегенова А.У., Антонов А.В. Проектный подход и планируемое обучение на основе онтологии // XII Междунар. Азиатская школа-семинар «Проблемы оптимизации сложных систем». - Новосибирск, 2016. - С. 306-312.

26 Тенденции развития компьютерных сетей и интернета // Журнал SkReview 05/2013 // URL: <http://arccn.ru/media/572>: 19.09.2015.

27 Сагадиев Е., Цифровизация позволит решить проблемы системы образования в РК // <http://profit.kz/news/45123/E-Sagadiev-cifrovizaciya-pozvolit-reshit-problemi-sistemi-obrazovaniya-v-RK/> : 20.03.2018.

28 Давлетбаева А.Р. Модель обеспечения качества дистанционного обучения на основе онтологического подхода // Образовательные технологии и общество. - 2015. - Т. 18, № 2. - С. 397-408.

29 Боровикова О.И., Загорюлько Ю.А. Организация порталов знаний на основе онтологий // Компьютерная лингвистика и интеллектуальные технологии: Труды междунар. семинара «Диалог'2002». - Протвино, 2002. - Т. 2. - С. 76-82.

30 Маликов А.В., Целиковский С. Модель системы дистанционного образования, основанная на онтологии предметных областей курсов обучения // Образовательные технологии и общество. - 2011. - Т. 14, № 3. - С. 387-394.

31 Гаврилина Е.А., Захаров М.А., Карпенко А.П., Смирнова Е.В. Онтологический подход к тестированию уровня владения обучающимся метапредметными понятиями // Наука и образование: научное издание МГТУ им. Н.Э. Баумана. - 2015. - № 2. - С. 136-149.

32 Дорохова О.Е. Семантические модели компетенций в адаптивной системе автоматизированного обучения // Современные проблемы науки и образования. - 2015. - № 3. - С. 434-543.

33 Загорюлько Ю.А. Автоматизация сбора онтологической информации об интернет-ресурсах для портала научных знаний // Известия Томского политехнического университета. - 2008. - Т. 312, № 5. - С. 114-119.

34 Кинтонова А.Ж., Рахижанова М.Б., Кинтонова А.Ж. Семантические технологии в дистанционном обучении // Новое слово в науке и практике. - 2014. - № 9. - С. 73-76.

35 Родзин С.И. Контекстно-зависимые мобильные обучающие системы // Известия ЮФУ. Технические науки. -2013. - № 7 (144). - С. 247-253.

36 Родзин С.И. M-learning - управление контентом в контекстно-зависимой мобильной системе обучения // Информатика, вычислительная техника и инженерное образование. - 2014. -№ 1 (16). - С. 53-61.

37 Андрианова С.В. Онтология в обучении англоязычной технической терминологии // Филология и лингвистика. - 2013. -№ 5 (68). - С. 100-104.

38 Белых С.Л. Онтология психологических понятий как инструмент управления знаниями в системе психологического // Когнитивный анализ и управление развитием ситуаций. Междунар. науч.-практ. мультikonф. «Управление большими системами - 2009». – М., 2009. - С. 182-187.

39 Сербин В.И. База знаний тренажерной системы как онтология предметной области // Информационные технологии в проектировании и производстве. - 2008. -№ 1. - С. 41-43.

40 Ужва А.Ю., Жукова И.Г., Кульцова М.Б. Онтологическая модель предметной области, обеспечивающая поддержку рассуждений по прецедентам

для поиска образовательных ресурсов // Известия Волгоградского государственного технического университета. - 2010. - Т. 9, № 11(71). - С. 110-113.

41 Гаврилина Е.А., Захаров М.А., Карпенко А.П., Смирнова Е.В. Онтологический подход к тестированию уровня владения обучающимся метапредметными понятиями // Наука и образование: научное издание МГТУ им. Н.Э. Баумана. - 2015. - № 2. - С. 136-149.

42 Дорохова О.Е. Семантические модели компетенций в адаптивной системе автоматизированного обучения // Современные проблемы науки и образования. - 2015. - № 3. - С. 434-543.

43 Лутошкина Н.В., Мурашова Л. М. Систематизация мультимедийного контента электронного курса на основе онтологии предметной области // Современные проблемы науки и образования. - 2013. - № 6. - С. 333-339.

44 Шполянская И.Ю., Мирошниченко И. И. Онтологическая модель информационно-образовательной среды учебного подразделения // Вестник Ростовского государственного экономического университета (РИНХ). - 2010. - № 31. - С. 201-209.

45 Карасюк В.В., Гвозденко М.В. Принципы построения и особенности функционирования интеллектуальной системы обучения // Вестник НТУ «ХПИ». - 2012. - № 38. - С. 77-83.

46 Gruber T.R. Translation Approach to Portable Ontology Specification. //Knowledge Acquisition. -1993.-№5/2.- P.199-220.

47 Челпанов И.В. Компетентностный подход при разработке государственных образовательных стандартов высшего кораблестроительного образования. - М.: Исследовательский центр проблем качества подготовки специалистов, 2005. - 97 с.

48 Чарнецки К., Айзенкер У. Порождающее программирование: методы, инструменты, применение. Для профессионалов. - СПб.: Питер, 2005.- 731 с

49 Субетто А.И. Интеграционная модель выпускника вуза на базе системодейательностного и компетентностного подхода. - Изд-во КГУ им. Н.А.Некрасова, Кострома, 2005. –28 с.

50 Субетто А.И. Онтология и феноменология педагогического мастерства. Книга первая. - Тольятти: Фонд «Развитие через образование», 1999. –208 с.

51 Правила организации учебного процесса по кредитной технологии обучения: утв. приказом Министра образования и науки Республики Казахстан от 20 апреля 2011 года № 152.

52 Хуторской А.В. Компетентностный подход в обучении. Научно-методическое пособие (Серия «Новые стандарты»). //М.: Издательство «Эйдос», -2013. - 73 с.

53 Criteria for Academic Bachelor's and Master's Curricula. - [http://www.jointquality.com/content/descriptors/AC\\_English\\_Gweb.pdf](http://www.jointquality.com/content/descriptors/AC_English_Gweb.pdf) : 12.06.2017.

54 Shared Dublin descriptors for Short Cycle, First Cycle, Second Cycle and Third Cycle Awards. - <http://www.dublin.ac.uk/>

//www.jointquality.com/content/descriptors/CompletesetDublinDescriptors.doc :  
28.06.2017.

55 Байденко В.И. Болонский процесс: проблемы, опыт, решения. Изд. 2-е, исправленное и дополненное. - М.: Исследовательский центр проблем качества подготовки специалистов, 2006. –111 с.

56 Байденко В.И. Болонский процесс: поиск общности европейских систем высшего образования (проект TUNING). - М.: Исследовательский центр проблем качества подготовки специалистов, 2006. –211 с.

57 Брауде Э. Технология разработки программного обеспечения. - СПб.: Питер, 2004.- 655 с.

58 Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на C++. /пер. с англ.- СПб.: Невский диалект, 1998.- 560 с.

59 Вигерс Карл. Разработка требований к программному обеспечению / пер. с англ. - М.: Изд. торговый дом "Русская Редакция", 2004.- 576 с.

60 Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. - СПб.: Питер, 2000. - 384 с.

61 Воинов А., Гаврилова Т.А., Данцин Е.Я., Язык визуального представления знаний и его место в САКЕ-технологии // Журнал Известия РАН, Теория и системы управления. -1996.-№2. - С.146-151.

62 Кроули Е.Ф., Программа CDIO: Описание целей и задач бакалаврского инженерного образования, Доклад CDIO №1. - изд. МИТ, 2001.// <http://www.cdio.org>: 18.02.2017.

63 Хассан Гома UML. Проектирование систем реального времени, параллельных и распределенных приложений / пер. с англ.-М.: ДМК Пресс, 2014.- 704 с.

64 Елина Е.Г., Фризен М.А. Образовательные технологии и методы высшей школы в США и странах Европы. // Образовательные технологии. - М.: Изд. дом "Народное образование", 2013. -№1.- С.31-37.

65 Ильин Г.Л. Современное мышление: от логики к аналогии. //Образовательные технологии. - М.: Изд. дом «Народное образование».- 2013. - №2. - С.21-33.

66 Коплиен Дж. Мультипарадигменное проектирование для C++. – Питер, 2005.- 235 с.

67 Котляров В.П. Основы тестирования программного обеспечения: Учебное пособие. - М.: Интернет-Университет Информационных технологий; Бином, Лаборатория Знаний, 2006.- 285 с.

68 Леонтьев А.Н. Деятельность. Сознание. Личность. - М.: Политиздат.- 1975. - 304 с.

69 Мейер Бертран. Почувствуй класс/Мейер Б./ пер.с.англ. под ред. В.А. Биллига.- М.: Национальный Открытый Университет "Интуит": БИНОМ. Лаборатория знаний, 2011.- 775 с.

70 Нариньяни А.С. Кентавр по имени ТЕОН: Тезаурус+Онтология //Труды Междунар. конф. - М.: ДИАЛОГ, 2001. - С.184-188.

71 Скопин И.А. Основы менеджмента программных проектов. Курс лекций: учебное пособие. - М.:ИНТУИТ.РУ «Интернет-Университет Информационных технологий», 2004.- 336 с.

72 Ньюмен Сэм. Создание микросервисов. - СПб.: Питер, 2016 .- 304 с.

73 Шмырева Н.А., Губанова М.И., Крецан З.В. Педагогические системы: научные основы, управление, перспективы развития. - Изд.: Общество с ограниченной ответственностью "Авторское издательство Кузбассвуиздат" (Кемерово), 2002. -99 с.

74 Белых С.Л. Онтология психологических понятий как инструмент управления знаниями в системе психологического образования // Когнитивный анализ и управление развитием ситуаций. Междунар. науч.-практ. мультikonф. «Управление большими системами - 2009». - Москва, 2009. - С. 182-187.

75 Загорулько Ю.А. Автоматизация сбора онтологической информации об интернет-ресурсах для портала научных знаний // Известия Томского политехнического университета. - 2008. - Т. 312, № 5. - С. 114-119.

76 Кафтанников И.Л., Коровин С.Е. Перспективы использования web-онтологий в учебном процессе // Образовательные технологии и общество. - 2003. - Т. 6, № 3. - С. 134-138.



## ПРИЛОЖЕНИЕ А

### Листинг программного интерпретатора

Для визуализации онтологий и характеристических моделей разработан программный интерпретатор, реализованный на объектно-ориентированном языке программирования C#.

В качестве средства разработки программного интерпретатора выражений знаний и для поддержки адаптивной образовательной среды было использовано передовое программное решение Microsoft Visual Studio 2013 - среда, позволяющая проектировать и создавать привлекательные приложения, удовлетворяющие самым взыскательным требованиям заказчиков.

По спецификации выражения знания, либо выражения спецификации характеристик опорного понятия собирается либо понятийный граф онтологии, либо характеристический граф.

#### **BaseClass.cs**

```
namespace WpfApplication1
{
    public class BaseClass<T>
    {
        public int ID { get; set; }
        public string Title { get; set; }
        public string SubTitle { get; set; }
        public int? ParentID { get; set; }
        public bool isRequierd { get; set; }
        public bool isAlternative { get; set; }
        public virtual List<T> Childs { get; set; }
        public string ConceptModel { get; set; }
    }
}
```

#### **Concept.cs**

```
namespace WpfApplication1
{
    public class Concept: BaseClass<Concept>
    {
        public override List<Concept> Childs { get; set; }
        public Concept() { }
        public Concept(int id, string title, string subtitle, string conceptstr)
        {
            base.Title = title;
            base.SubTitle = subtitle;
            base.ID = id;
            base.ConceptModel = conceptstr;
        }
    }
}
```

```

public Concept(string concept)
{
    try { if ((concept.ToCharArray())[0] == '*')
        { base.isRequierd = true; }
        if ((concept.ToCharArray())[0] == '+')
            { base.isRequierd = false; }
        base.Title = Convert.ToString((concept.ToCharArray())[1]);
        base.SubTitle = concept.Substring(2, concept.IndexOf('.')-2);
        base.ID = Convert.ToInt16(concept.Substring(concept.IndexOf('.') + 1));
    }
    catch (Exception)
    { throw new ParseException("Ошибка в входной строке"); }
}
public Concept AddChild(string concept, Concept Parent, bool isAlternative =
false)
{ if (this.Childs == null) { this.Childs = new List<Concept>(); }
  Concept childconcept = new Concept();
  if (concept[0] == '*')
  {
    if (!Parent.isRequierd) throw new ParseException("Необязательное опорное
понятие не может иметь обязательных понятий");
    childconcept.isRequierd = true;
  }
  if (concept[0] == '+')
  { childconcept.isRequierd = false; }
  childconcept.Title = Convert.ToString((concept.ToCharArray())[1]);
  childconcept.ID = Convert.ToInt16(concept.Substring(2));
  childconcept.ParentID = Parent.ID;
  childconcept.isAlternative = isAlternative;
  this.Childs.Add(childconcept);
  return childconcept;
}
}
}
}

```

### **Feature.cs**

```

namespace WpfApplication1
{
    public class Concept: BaseClass<Concept>
    {
        public override List<Concept> Childs { get; set; }

        public Concept() { }
        public Concept(int id, string title, string subtitle, string conceptstr)
        {

```

```

base.Title = title;
base.SubTitle = subtitle;
base.ID = id;
base.ConceptModel = conceptstr;
}
public Concept(string concept)
{ try {
if ((concept.ToCharArray())[0] == '*')
{ base.isRequierd = true; }
if ((concept.ToCharArray())[0] == '+')
{ base.isRequierd = false; }
base.Title = Convert.ToString((concept.ToCharArray())[1]);
base.SubTitle = concept.Substring(2, concept.IndexOf('.')-2);
base.ID = Convert.ToInt16(concept.Substring(concept.IndexOf('.') + 1));
}
catch (Exception)
{ throw new ParseException("Ошибка в входной строке"); }
}
public Concept AddChild(string concept, Concept Parent, bool isAlternative =
false)
{
if (this.Childs == null)
{ this.Childs = new List<Concept>(); }
Concept childconcept = new Concept();
if (concept[0] == '*')
{
if (!Parent.isRequierd) throw new ParseException("Необязательное опорное
понятие не может иметь обязательных понятий");
childconcept.isRequierd = true;
}
if (concept[0] == '+') { childconcept.isRequierd = false; }
childconcept.Title = Convert.ToString((concept.ToCharArray())[1]);
childconcept.ID = Convert.ToInt16(concept.Substring(2));
childconcept.ParentID = Parent.ID;
childconcept.isAlternative = isAlternative;
this.Childs.Add(childconcept);
return childconcept;
}
}
}

```

### **Knowledge.cs**

```

namespace WpfApplication1
{
public class Knowledge

```

```

{
public List<Concept> Concepts { get; set; }
public List<Feature> Features { get; set; }
private int IdSequence = 1;
public Knowledge()
{
this.Concepts = new List<Concept>();
this.Features = new List<Feature>();
}
private List<Concept> ParseConceptStr(string str)
{
List<Concept> strList = new List<Concept>();
List<Concept> LastAdded = new List<Concept>();
int index = 0;
bool start = false;
int startindex = 0;
bool innerpart = false;
bool alternative = false;
if ((str[0] != '*') && (str[0] != '+')) throw new ParseException("Понятие должно
начинаться с символа '*' или '+'");
foreach (var ch in str.ToCharArray())
{
if (((ch == '*') || (ch == '+') || (ch == '~') || (ch == '(') || (ch == ')') || (ch == ';'))) &&
(start))
{ if (innerpart)
{
if (!strList[strList.Count - 1].IsRequired) throw new
ParseException("Необязательное опорное понятие не может иметь
конкретизирующих понятий");
var item = LastAdded[LastAdded.Count -
1].AddChild(str.Substring(startindex, index - startindex), strList[strList.Count -
1], alternative);
alternative = false;
if (ch == '(')
{ if (LastAdded.Count == 1)
{ throw new ParseException("Превышено количество уровней
конкретизации понятия. Допускается не более двух."); }
LastAdded.Add(item);
}
}
else
{
var newconcept = new Concept(str.Substring(startindex, index - startindex));
if (alternative == true) newconcept.IsAlternative = true;
}
}
}
}
}
}

```

```

        alternative = false;
        str.Add(newconcept);
        if (ch == '(')
        {
            if (str[index + 1] == ')') throw new ParseException("Множество
конкретизирующих понятий (*C1~+C2) не может быть пустым");
            if ((str[index+1] != '*') && (str[index+1] != '+')) throw new
ParseException("Понятие должно начинаться с символа '*' или '+'");
            innerpart = true;
            LastAdded.Add(newconcept);
        }
        if(ch=='~')
        { alternative = true; }
    }
    if (ch == ')')
    { LastAdded.RemoveAt(LastAdded.Count - 1);
    if (LastAdded.Count==0)
    {
        if ((ch == ')') && (str[index + 1] != '*') && (str[index + 1] != '+') &&
(str[index + 1] != '~') && (str[index + 1] != ';')) throw new ParseException("Понятие
должно начинаться с символа '*' или '+'");
        innerpart = false;
    }
    }
    if (ch == '~')
    { alternative = true; }
    start = false;
}
if ((ch=='') &&(!start) && (innerpart)&&(str[index-1]=='))
{
    LastAdded.RemoveAt(LastAdded.Count - 1);
    if (LastAdded.Count==0)
    { innerpart = false; }
}
if ((ch == '*') || (ch == '+'))
{ startindex = index; start = true; }
if ((ch == '~') && (str[index - 1]=='))
{ alternative = true; }
index++;
}
return str;
}
private List<Feature> ParseFeatureStr(string str)
{

```

```

List<Feature> strс = new List<Feature>();
List<Feature> LastAdded = new List<Feature>();
int index = 0;
bool start = false;
int startindex = 0;
bool innerpart = false;
bool alternative = false;
if ((str[0] != '*') && (str[0] != '+')) throw new ParseException("Понятие должно
начинаться с символа '*' или '+'");
foreach (var ch in str.ToCharArray())
{
    if (((ch == '*') || (ch == '+') || (ch == '~') || (ch == '(') || (ch == ')') || (ch == ';'))) &&
(start))
    {
        if (innerpart)
        {
            if (!strс[strс.Count - 1].isRequierd) throw new
ParseException("Необязательное опорное понятие не может иметь
конкретизирующих понятий");
            var item = LastAdded[LastAdded.Count -
1].AddChild(str.Substring(startindex, index - startindex), strс[strс.Count - 1],
alternative);
            alternative = false;
            if (ch == '(')
            { if (LastAdded.Count == 1)
            {
                throw new ParseException("Превышено количество уровней
конкретизации понятия. Допускается не более двух.");
            }
            LastAdded.Add(item);
            }
        }
        else
        {
            var newconcept = new Feature(str.Substring(startindex, index - startindex));
            if (alternative == true) newconcept.isAlternative = true;
            alternative = false;
            strс.Add(newconcept);
            if (ch == '(')
            {
                if (str[index + 1] == ')') throw new ParseException("Множество
конкретизирующих понятий (*C1~+C2) не может быть пустым");
                if ((str[index + 1] != '*') && (str[index + 1] != '+')) throw new
ParseException("Понятие должно начинаться с символа '*' или '+'");
            }
        }
    }
}

```

```

        innerpart = true;
        LastAdded.Add(newconcept);
    }
    if (ch == '~')
    { alternative = true; }
    }
    if (ch == ')')
    {
        LastAdded.RemoveAt(LastAdded.Count - 1);
        if (LastAdded.Count == 0)
        {
            if ((ch == ')') && (str[index + 1] != '*') && (str[index + 1] != '+') &&
(str[index + 1] != '~') && (str[index + 1] != ';')) throw new ParseException("Понятие
должно начинаться с символа '*' или '+'");
            innerpart = false;
        }
    }
    if (ch == '~')
    { alternative = true; }
    start = false;
}
if ((ch == ')') && (!start) && (innerpart) && (str[index - 1] == '))
{
    LastAdded.RemoveAt(LastAdded.Count - 1);
    if (LastAdded.Count == 0)
    { innerpart = false; }
}
if ((ch == '*') || (ch == '+'))
{ startindex = index; start = true; }
if ((ch == '~') && (str[index - 1] == '))
{ alternative = true; }
index++;
}
return str;
}
public void AddConcept(string concept)
{
    if ((concept.IndexOf("<=") == -1) || (concept[concept.Length-1]!=';'))
    {
        throw new ParseException("Не найден обязательный символ ';' или '<='");
    }
    else
    {
        string title = concept.Substring(0, 1);

```

```

        string subtitle = concept.Substring(1, concept.IndexOf("<=")-1);
        string conceptstr = concept.Substring(concept.IndexOf("<=") + 2);
        Concept newconcept = new Concept(IdSequence, title, subtitle,concept)
    {isRequierd = true};
        newconcept.Childs = new List<Concept>();
        newconcept.Childs.AddRange(ParseConceptStr(conceptstr).AsEnumerable());
        this.Concepts.Add(newconcept);
    }
}
public void AddFeature(string feature)
{
    if ((feature.IndexOf("<=") == -1) || (feature[feature.Length - 1] != ';'))
    { throw new ParseException("Не найден обязательный символ ';' или '<='"); }
    else
    {
        string title = feature.Substring(0, 1);
        string subtitle = feature.Substring(1, feature.IndexOf("<=")-1);
        string feauterstr = feature.Substring(feature.IndexOf("<=") + 2);
        Feature newconcept = new Feature(IdSequence, title, subtitle, feature) {
isRequierd = true };
        newconcept.Childs = new List<Feature>();
        newconcept.Childs.AddRange(ParseFeatureStr(feauterstr).AsEnumerable());
        this.Features.Add(newconcept);
    }
}
}
public class ParseException : Exception
{
    private string _message;
    public override string Message
    { get { return _message; } }
    public ParseException(string message)
    { this._message = message; }
}
}

```

### **Visualizer.cs**

```

namespace WpfApplication1
{
    class Visualizer<T> where T : BaseClass<T>
    {
        Canvas _Canvas;
        BitmapImage _Bitmap;
        public BitmapImage GetVisualModel(T content)
        {

```



```

    _Canvas = new Canvas();
    _Canvas.Width = 600;
    _Canvas.Height = 350;
    _Canvas.Background = Brushes.White;
    DrawModel(content);
    return CanvasToBitmap(_Canvas);
}
private void DrawModel(T content)
{
    _Canvas.Children.Clear();
    int top = 20;
    Border border = new Border() { BorderThickness = new Thickness(1),
    BorderBrush = Brushes.Black, Height = 25, Width = 60 };
    TextBlock tbx = new TextBlock() { FontSize = 18, Text = content.Title +
    content.SubTitle, HorizontalAlignment = HorizontalAlignment.Center };
    border.Child = tbx;
    // Line ln = new Line() { X1 = _Canvas.Width / 2, X2 = _Canvas.Width / 2, Y1
    = 0, Y2 = 500, Stroke = Brushes.Red, StrokeThickness = 1 };
    //_Canvas.Children.Add(ln);
    Canvas.SetLeft(border, _Canvas.Width / 2 - 25);
    Canvas.SetTop(border, top);
    _Canvas.Children.Add(border);
    DrawChildModel(content, top + 105, _Canvas.Width / 2, _Canvas.Width);
    _Canvas.UpdateLayout();
}
private void DrawChildModel(T content, int top, double parentcenter, double
desiredsize)
{
    double range = desiredsize / (content.Chlds.Count);
    double offset = parentcenter - (desiredsize / 2);
    double borderleft = range / 2 + offset;
    Line LastLine = null;
    foreach (var item in content.Chlds)
    {
        Border border = new Border() { BorderThickness = new Thickness(0), Height =
        25, Width = 40 };
        TextBlock tbx = new TextBlock() { Margin = new Thickness(3, 5, 0, 0),
        FontSize = 14, Text = item.Title + (item.SubTitle == null ? "" : item.SubTitle+".") +
        item.ID, HorizontalAlignment = HorizontalAlignment.Left };
        border.Child = tbx;
        Canvas.SetLeft(border, borderleft + 4);
        Canvas.SetTop(border, top - 12);
        _Canvas.Children.Add(border);
    }
}

```

```

        Line line = new Line() { X1 = parentcenter, X2 = borderleft, Y1 = top - 80, Y2
= top - 5, Stroke = Brushes.Black, StrokeThickness = 1 };
        if (item.isAlternative)
        {
            double X1 = LastLine.X1 + 34 * (LastLine.X2 - LastLine.X1) /
Math.Sqrt(Math.Pow(LastLine.X2 - LastLine.X1, 2) + Math.Pow(LastLine.Y2 -
LastLine.Y1, 2));
            double Y1 = LastLine.Y1 + 34 * (LastLine.Y2 - LastLine.Y1) /
Math.Sqrt(Math.Pow(LastLine.X2 - LastLine.X1, 2) + Math.Pow(LastLine.Y2 -
LastLine.Y1, 2));
            double X2 = line.X1 + 34 * (line.X2 - line.X1) / Math.Sqrt(Math.Pow(line.X2 -
line.X1, 2) + Math.Pow(line.Y2 - line.Y1, 2));
            double Y2 = line.Y1 + 34 * (line.Y2 - line.Y1) / Math.Sqrt(Math.Pow(line.X2 -
line.X1, 2) + Math.Pow(line.Y2 - line.Y1, 2));
            // Line ln = new Line() { X1 = X1, X2 = X2, Y1 = Y1, Y2 = Y2, Stroke =
Brushes.Black, StrokeThickness = 1 };
            // _Canvas.Children.Add(ln);
            System.Windows.Shapes.Path pth = new System.Windows.Shapes.Path() {
Stroke = Brushes.Black, StrokeThickness = 1 };
            pth = DrawArc(new Point(X1, Y1), new Point(X2, Y2), parentcenter, top - 80);
            Canvas.SetTop(pth, top - 80);
            Canvas.SetLeft(pth, parentcenter);
            _Canvas.Children.Add(pth);
        }
        LastLine = line;
        _Canvas.Children.Add(line);
        Ellipse circle = new Ellipse() { Width = 10, Height = 10, StrokeThickness = 1,
Stroke = Brushes.Black, Fill = Brushes.Transparent };
        if (item.isRequierd)
        { circle.Fill = Brushes.Black; }
        Canvas.SetTop(circle, top - 5);
        Canvas.SetLeft(circle, borderleft - 5);
        _Canvas.Children.Add(circle);
        if (item.Childs != null)
        { DrawChildModel(item, top + 80, borderleft, range); }
        borderleft += range;
    }
}
System.Windows.Shapes.Path DrawArc(Point p1, Point p2, double leftoffset,
double toppffset)
{
    int R = 33; double X1 = 0; double Y1 = 0; int x = 0; int y = R;
    int delta = 1 - 2 * R; int error = 0; string pathIII = ""; string pathIV = "";
    bool first = true; bool first1 = true;

```

```

while (y >= 0)
{
//drawpixel(X1 + x, Y1 + y) IV
//drawpixel(X1 - x, Y1 + y) III
if (first)
{
if (((X1 - x + leftoffset) >= p1.X) && ((X1 - x + leftoffset) <= p2.X))
{ pathIII += "M" + (X1 - x) + "," + (Y1 + y) + " "; first = false; }
}
else
{
if (((X1 - x + leftoffset) >= p1.X) && ((X1 - x + leftoffset) <= p2.X))
{ pathIII += "L" + (X1 - x) + "," + (Y1 + y) + " "; }
}
if (first1)
{
if (((X1 + x + leftoffset) >= p1.X) && ((X1 + x + leftoffset) <= p2.X))
{ pathIV += "M" + (X1 + x) + "," + (Y1 + y) + " "; first1 = false; }
}
else
{
if (((X1 + x + leftoffset) >= p1.X) && ((X1 + x + leftoffset) <= p2.X))
{ pathIV += "L" + (X1 + x) + "," + (Y1 + y) + " "; }
}
//(((X1 - x + leftoffset) >= p1.X) && ((X1 - x + leftoffset) <= p2.X))
//pathIV += "M" + (X1 - x) + "," + (Y1 + y) + " ";
error = 2 * (delta + y) - 1;
if ((delta < 0) && (error <= 0))
{ delta += 2 * ++x + 1; continue; }
error = 2 * (delta - x) - 1;
if ((delta > 0) && (error > 0))
{ delta += 1 - 2 * --y; continue; }
x++;
delta += 2 * (x - y);
y--;
}
System.Windows.Shapes.Path pth = new System.Windows.Shapes.Path() {
Stroke = Brushes.Black, StrokeThickness = 1 };
pth.Data = Geometry.Parse(pathIII + pathIV);
return pth;
}
private BitmapImage CanvasToBitmap(Canvas surface)
{ Transform transform = surface.LayoutTransform;
surface.LayoutTransform = null;

```

```

Size size = new Size(surface.Width, surface.Height);
surface.Measure(size);
surface.Arrange(new Rect(size));
RenderTargetBitmap renderBitmap = new RenderTargetBitmap(
    (int)size.Width, (int)size.Height, 96d, 96d, PixelFormats.Pbgra32);
renderBitmap.Render(surface);
var bitmapImage = new BitmapImage();
var bitmapEncoder = new PngBitmapEncoder();
bitmapEncoder.Frames.Add(BitmapFrame.Create(renderBitmap));
using (var stream = new MemoryStream())
{
    bitmapEncoder.Save(stream);
    stream.Seek(0, SeekOrigin.Begin);
    bitmapImage.BeginInit();
    bitmapImage.CacheOption = BitmapCacheOption.OnLoad;
    bitmapImage.StreamSource = stream;
    bitmapImage.EndInit();
}
_Bitmap = bitmapImage;
return bitmapImage;
}
public void SaveToFile(string path)
{
    PngBitmapEncoder encoder = new PngBitmapEncoder();
    encoder.Frames.Add(BitmapFrame.Create((BitmapImage)_Bitmap));
    using (var filestream = new FileStream(path, FileMode.Create))
        encoder.Save(filestream);
}
}
}
...

```

## ПРИЛОЖЕНИЕ Б

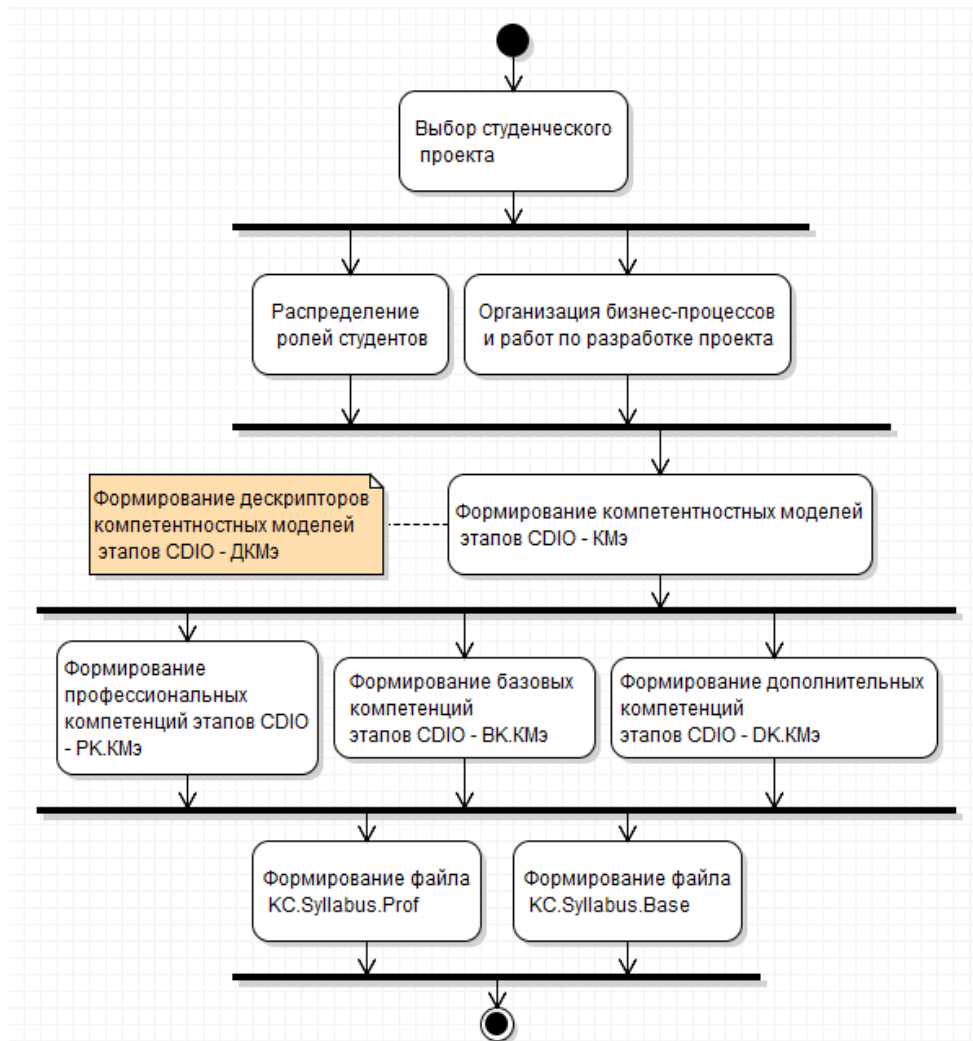


Рисунок Б.1 -Сценарий работы эксперта со средой

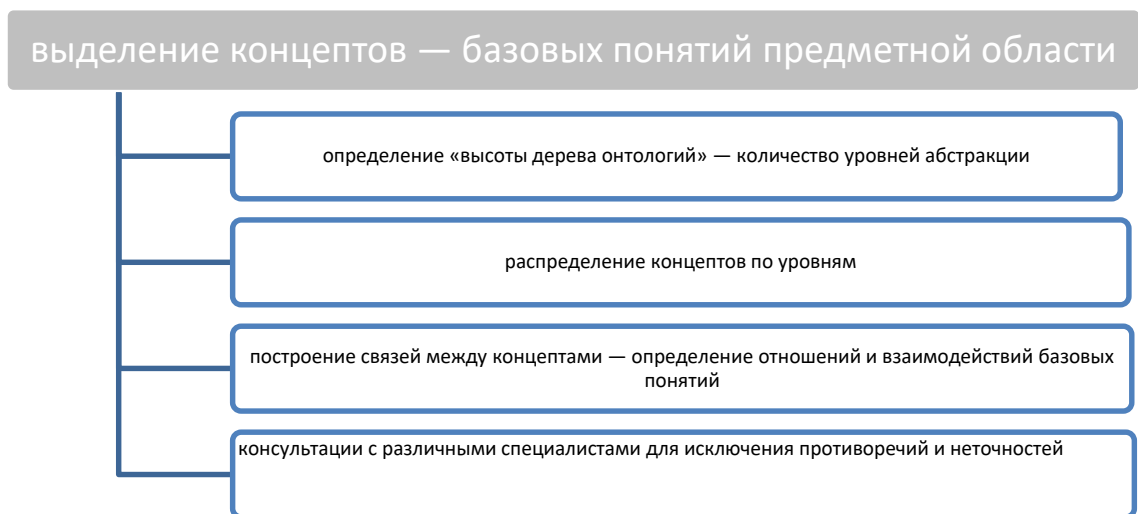


Рисунок Б.2 - Базовый алгоритм онтологического инжиниринга



Рисунок Б.3 - Многоуровневая система подготовки специалистов



Рисунок Б.4 - Многоуровневая система формирования онтологий



Рисунок Б.5 - Четырехслойная интегрированная онтология обучения