

Санкт-Петербургский государственный университет

Математическое обеспечение и администрирование информационных систем

Кафедра системного программирования

Добряков Дмитрий Алексеевич

# Разработка системы создания курсов адаптивного обучения

Бакалаврская работа

Научный руководитель:  
д. ф.-м. н., профессор Терехов А. Н.

Научный консультант:  
ст. преп. Немешев М. Х.

Рецензент:  
к. ф.-м. н., доцент Костин В. А.

Санкт-Петербург  
2020

SAINT-PETERSBURG STATE UNIVERSITY

Software and administration of information systems  
System Engineering

Dmitry Dobryakov

# Development of an adaptive learning courses creation system

Graduation Thesis

Scientific supervisor:  
professor Terekhov A. N.

Scientific consultant:  
senior lecturer Nemeshev M. H.

Reviewer:  
associate professor Kostin V. A.

Saint-Petersburg  
2020

# Оглавление

<b>Введение</b>	<b>4</b>
<b>1. Постановка задачи</b>	<b>5</b>
<b>2. Обзор</b>	<b>6</b>
2.1. Адаптивное обучение . . . . .	6
2.2. Адаптивное тестирование . . . . .	7
2.3. Существующие разработки . . . . .	8
<b>3. Реализация</b>	<b>10</b>
3.1. Архитектура и основные идеи . . . . .	10
3.2. Генетический алгоритм . . . . .	12
3.3. Адаптивное тестирование . . . . .	14
3.3.1. Адаптация сложности модулей . . . . .	15
3.4. Организация данных . . . . .	16
3.4.1. Comapping API . . . . .	16
3.4.2. Jsonbin API . . . . .	19
<b>4. Эксперименты</b>	<b>20</b>
<b>Заключение</b>	<b>22</b>
<b>Список литературы</b>	<b>23</b>

# Введение

В настоящее время сфера онлайн-образования стремительно развивается и с каждым годом становится все более востребованной. Многие университеты постепенно движутся в сторону дистанционного подхода к обучению, и появляется все больше массовых открытых онлайн-курсов. Доступность учебных материалов и гибкость в форматах их подачи делают процесс обучения более комфортным и зачастую более эффективным в сравнении с традиционным способом.

В то же время, с ростом популярности онлайн-образования и массовости аудитории слушателей онлайн-курсов, становится как никогда актуальным вопрос построения индивидуальной образовательной траектории и персонализации обучающего контента. Люди, проходящие один курс, отличаются друг от друга как по своим начальным знаниям и способностям, так и по ряду прочих характеристик, включая память, мотивацию, преобладающий тип восприятия информации и другие. Правильный их учет может значительно повысить скорость и эффективность усвоения материала.

В связи с этим возникает задача построения таких электронных образовательных систем, которые смогут самостоятельно обеспечить индивидуальный подход к особенностям каждого обучающегося и адаптироваться под его изменяющийся уровень знаний, в некоторой степени взяв на себя роль реального преподавателя. Такие системы называются системами адаптивного обучения и для реализации его принципов должны уметь:

1. оценивать индивидуальные качества обучаемого, его входной уровень подготовки;
2. базируясь на этой динамически-обновляемой оценке, предоставлять персональный план обучения.

Системы адаптивного обучения повышают эффективность и скорость обучения и снимают риски наличия неосознанной некомпетенции, обеспечивая гибкий учебный процесс.

# 1. Постановка задачи

Целью данной работы является разработка системы, предоставляющей возможности для создания и прохождения курсов, реализующих адаптивное обучение, в связи с чем сформулированы следующие задачи:

- Исследовать существующие разработки и алгоритмы в данной области;
- Разработать архитектуру приложения;
- Подобрать и реализовать алгоритмы адаптивного обучения:
  - алгоритм поиска индивидуальной образовательной траектории;
  - метод адаптивного тестирования;
- Реализовать систему с использованием выбранных алгоритмов;
- Построить при помощи системы адаптивный курс и проанализировать результаты его прохождения реальными людьми.

## 2. Обзор

### 2.1. Адаптивное обучение

Систему адаптивного обучения традиционно принято делить на следующие компоненты [9]:

- Экспертная модель: содержит всю информацию о курсе, включая структуру, учебные материалы, задания для проверки и дополнительную мета-информацию, которая в дальнейшем может быть использована для обеспечения адаптивности.
- Модель студента: содержит данные о студенте, его текущем прогрессе и степени усвоения пройденного материала, индивидуальных характеристиках.
- Учебная модель: предоставляет студенту актуальную обучающую информацию, определяет алгоритмы для обработки результатов его тестирования и выявления сильных и слабых сторон.
- Учебная среда: пользовательский интерфейс для работы с системой.

Адаптивность курса означает, что его видимое содержание подстраивается и изменяется с течением времени, оценивая изменения в показателях обучаемого. Для того, чтобы адаптивный подход был осуществим, необходимо, чтобы материал курса был поделен на некрупные части — элементарные модули или объекты обучения [1], которые в некоторых случаях можно переставлять между собой для организации персональной последовательности обучения.

С учетом наличия логических взаимосвязей между элементарными модулями (например, один модуль можно изучать только после завершения другого), необходимо также определить некоторые правила, накладывающие ограничения на возможные последовательности их порядка.

Построение индивидуальной траектории обучающих модулей – основная задача адаптивной обучающей системы. Для ее решения необходимо располагать знаниями о студенте, проходящем учебный курс. В первую очередь, требуется иметь некоторую оценку его прогресса и изменяющегося уровня знаний. Также полезно обладать какой-то информацией о его индивидуальных особенностях, понимать, какие темы для него более сложны, конечные цели обучения и так далее.

Для оценки уровня знаний студента применимы методы адаптивного тестирования.

## 2.2. Адаптивное тестирование

Адаптивное тестирование (САТ = Computerized Adaptive Testing) предполагает динамическое формирование списка предлагаемых заданий, основанное на полученных ответах.

Конечная цель адаптивного тестирования — количественно оценить уровень знаний студента. Процесс можно разделить на следующие задачи (этапы 2-3 повторяются до выполнения 4):

- Критерий начала тестирования;
- Определение следующего задания (с учетом текущего состояния уровня знаний);
- Корректировка уровня знаний (в результате анализа данного ответа);
- Критерий завершения тестирования.

На практике для реализации этих задач используются теоретически-обоснованные процедуры с набором параметров, определяющих требуемую точность оценки. Основными моментами при разработке САТ являются [2]:

- модель ответа (Response Model), описывающая ответы студентов на конкретный тест в зависимости от их уровня знаний;

- резерв тестов (Item pool) — набор откалиброванных по различным уровням тестов, из которого впоследствии выбираются наиболее подходящие для конкретного студента.

Для выполнения поставленных задач часто применяется Item Response Theory – психометрическая теория, основанная на вероятностном моделировании результатов тестирования. Вероятность того, что студент с определенной оценкой уровня знаний правильно ответит на вопрос, определяется с помощью функции, которая называется Item Characteristic Curve (ICC) и калибруется для каждого задания. Методы калибровки кривой ICC являются предметом отдельного исследования и рассматриваются, к примеру, в [8].

Существуют различные вариации кривой ICC, принимающие разные параметры, такие как, например, фактор угадывания — вероятность правильного ответа при отсутствии знаний, или сложность, определяющаяся уровнем знаний, при котором вероятности правильного и неправильного ответа совпадают.

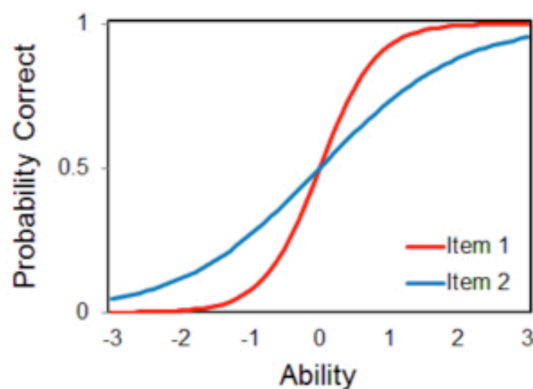


Рис. 1: Пример кривых ICC

### 2.3. Существующие разработки

В настоящее время исследования в области адаптивного обучения активно проводятся с нарастающей частотой. Были предложены разные подходы к построению индивидуальной образовательной траектории, основанные на приближенных методах.



Так, например, группа испанских исследователей [6], сведя данную проблему к перестановочной задаче удовлетворения ограничений (PermutCSP), предложила использовать генетический алгоритм для построения персональной последовательности обучающих курсов.

Ими же (немного в другом составе) [4] до этого был предложен подход, основанный на методе роя частиц вместо генетического алгоритма.

Исследователи из Томского государственного университета систем управления [13] описали модель адаптивного курса, основанную на понятии компетенций и кривой забывания, и также использовали генетический алгоритм для генерации траектории обучения.

А на базе Оренбургского государственного университета было проведено исследование [11] о применении генетического алгоритма к задаче адаптации материала под преобладающий тип восприятия информации.

К готовым системам адаптивного обучения можно отнести, например, Area9 Rhapsode<sup>1</sup>, где обучение происходит за счет ответов на вопросы с указанием уверенности в ответе. Эта система ориентирована на повышение профессиональной квалификации работников, и разработкой курса на заказ занимается специальная команда.

Также есть Knewton<sup>2</sup> – первая в мире система адаптивного обучения, которая уже внедрялась в процесс образования в качестве эксперимента и по его итогам улучшила результаты студентов Аризонского государственного университета на 18% и снизила количество отчислений на практически 50% [10].

---

<sup>1</sup><https://area9lyceum.com/rhapsode/>

<sup>2</sup><https://knewton.com/>

## 3. Реализация

### 3.1. Архитектура и основные идеи

Система реализована в виде мобильного приложения для платформы iOS с использованием языка Swift. В решении используется несколько основных видов сущностей, взаимодействующих между собой:

- **Module (Модуль)**  
Модуль представляет собой некоторый блок материалов курса. Курс состоит из структурированного набора модулей, и его прохождение включает в себя предъявление наиболее подходящих модулей в нужном порядке.
- **Competence (Компетенция)**  
Компетенция – это некоторый набор навыков или умений, получаемый студентом в результате прохождения модулей или являющийся частью требований, необходимых чтобы приступить к определенному модулю.
- **Student (Студент)**  
Эта сущность представляет собой модель студента, проходящего курс, и хранит в себе необходимую информацию для обеспечения адаптивности обучения, включая историю пройденных модулей и текущий уровень освоения компетенций.
- **TestItem (Тестовое задание)**  
Тестовые задания используются для проверки усвоения студентом пройденных модулей и корректировки модели студента на основе его ответов.

Адаптивный курс состоит из структурированного в виде дерева набора обучающих модулей и дерева компетенций, полученных в ходе декомпозиции целей обучения. Тесты также относятся к дереву компетенций и привязываются к ним в качестве листьев.

Каждый модуль снабжается набором входных (необходимых для освоения перед тем, как приступить) и выходных (получаемых в результате) компетенций. Считается, что модуль может давать любую компетенцию только в полном объеме (если это не так, то такую компетенцию можно разбить на несколько субкомпетенций в дереве). Помимо компетенций, в модуле также содержится информация о его сложности (которая может адаптироваться для разных студентов) и рассчитанной продолжительности.

Компетенции снабжаются набором тестов, призванных измерить их понимание студентом в процессе обучения для обеспечения адаптивности.

На рисунке 2 изображена диаграмма классов, отражающая архитектуру решения.

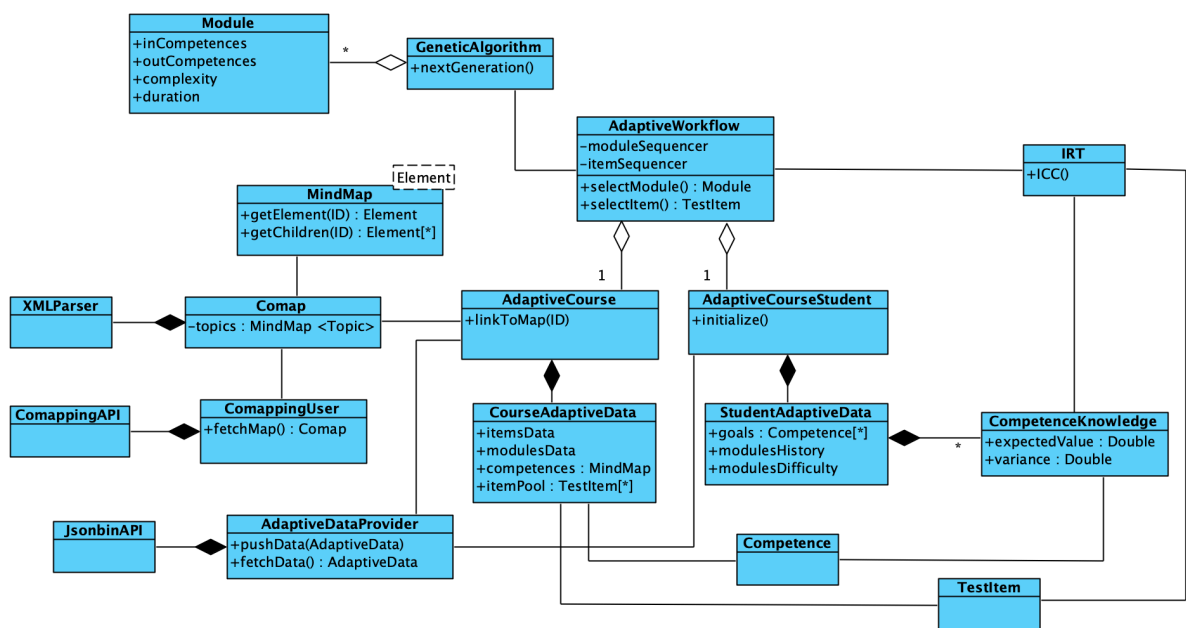


Рис. 2: UML-диаграмма классов

Классы AdaptiveCourse и AdaptiveCourseStudent отвечают за управление адаптивными данными, связанными соответственно с курсом и со студентом, проходящим курс. Оба класса занимаются получением и отправкой данных на сервер, а также их кэшированием.

Класс AdaptiveWorkflow отвечает за взаимодействие студента с материалами курса. Распоряжаясь ссылками на объекты AdaptiveCourse

и `AdaptiveCourseStudent`, он передает данные в алгоритмы для формирования траектории и проведения адаптивного тестирования.

За построение индивидуальной последовательности модулей отвечает класс `ModuleSequencer`, использующий `GeneticAlgorithm` в качестве алгоритмической основы, в то время как класс `ItemSequencer` занимается выборкой тестовых заданий.

Хранение модулей, компетенций и тестов курса в виде дерева осуществляется при помощи структуры `MindMap`, предоставляющей множество методов для удобного оперирования узлами и их индексирования по уникальному идентификатору.

## 3.2. Генетический алгоритм

Для реализации поиска индивидуальной образовательной траектории был избран генетический алгоритм на основе принципов, описанных в [6].

Из всех модулей курса на основании целевых компетенций студента выбирается набор, который потенциально может участвовать в искомой траектории, после чего каждому из модулей сопоставляется идентифицирующее его натуральное число. Таким образом, перестановка натуральных чисел длины, равной количеству модулей, определяет одну из возможных траекторий обучения. Она же является конкретной особью в генетическом алгоритме.

В качестве мутации используется перестановка местами двух произвольных элементов особи.

Отбор особей для скрещивания производится с помощью метода турнирной селекции. Для скрещивания используется метод `order crossover` (рис. 3), который зачастую применяется в реализациях генетического алгоритма, в которых важен порядок. Из рисунка видно, что этот метод сохраняет в потомке блок одного из предков, при этом перенося порядок всех оставшихся элементов согласно второму предку.

Для гарантии качественных результатов в алгоритме реализована стратегия элитизма, сохраняющая из каждого поколения определенное

количество наиболее приспособленных особей.

```
Parent 1: 8 4 7 3 6 2 5 1 9 0
Parent 2: 0 1 2 3 4 5 6 7 8 9
Child 1:  0 4 7 3 6 2 5 1 8 9
```

Рис. 3: Метод скрещивания: order crossover

Во избежание ранней сходимости к единому представителю и обеспечения вариативности популяции, в случае возникновения дубликатов осуществляется форсированная мутация. А с учетом особой важности самого первого элемента в искомой последовательности модулей, был также реализован специальный вид мутации, который в качестве одного из генов для перестановки всегда выбирает первый. Этот вид мутации производится только в тех случаях, когда мутированная особь дает лучший показатель приспособленности, и значительно снижает шансы получения некачественного первого модуля.

Функция приспособленности (или функция оценки) является, пожалуй, наиболее важной частью в генетическом алгоритме. Именно она определяет эффективность и работоспособность всего алгоритма, поэтому очень важно правильно ее подобрать.

В текущей реализации системы для оценки качества траектории разработан следующий подход.

1. Прохождение измеряемой траектории моделируется виртуальным студентом, при этом на каждом шаге отслеживаются "изученные" компетенции, "пропущенные" компетенции (которые были использованы как требование к модулю, но не были изучены на тот момент) и "просмотренные" компетенции (те, что были получены в результате некоторого модуля, входные требования к которому не были выполнены). Полученные значения суммируются с различными коэффициентами для формирования общей накопленной ошибки до тех пор, пока все целевые компетенции студента не будут выполнены.

2. Для сортировки траекторий с одинаковым значением ошибки измеряются общие усилия, затраченные моделируемым студентом на прохождение траектории, рассчитанные из времени и сложности модулей. Эта величина берется со значительно меньшим коэффициентом относительно накопленной ошибки, чтобы исключить формирование неправильных траекторий, когда это возможно.

Описанный метод протестирован на различных начальных условиях и показывает результаты, соответствующие логическим ожиданиям, за допустимое время.

### 3.3. Адаптивное тестирование

Для тестирования студента и оценки освоения им компетенций, представленных в курсе, используются вероятностные алгоритмы из Item Response Theory. Каждое тестовое задание снабжается своей Item Characteristic Curve, описывающей вероятность правильно ответить на вопрос при известном уровне тестируемого. В качестве основы ИСС используется логистическая модель с тремя параметрами, в оригинале описываемая следующей формулой:

$$p(\theta) = c + \frac{1 - c}{1 + e^{-a(\theta - b)}} \quad (1)$$

Параметр  $c$  здесь соответствует фактору угадывания, то есть вероятности случайно ответить на вопрос правильно.  $b$  означает сложность задания, что более точно выражается как тот уровень знаний, при котором вероятность ответить соответствует 50%. Параметр  $a$  отражает, насколько резко меняются шансы ответить верно с изменением знаний студента.

Одно не слишком существенное различие описываемой реализации с формулой (1) состоит в области определения этой функции. Если оригинал определяется для всей вещественной оси, то с практической точки зрения в ходе работы показалось удобным определить нижнюю и верхнюю границы возможных значений компетенций и установить область определения в рамках отрезка  $[0, 1]$ . Для того чтобы это сделать,

используется специально подобранная биекция  $[-\infty, \infty] \rightarrow [0, 1]$ , заданная формулой  $f(x) = \frac{e^{4x}}{1+e^{4x}}$  и обладающая производной в нуле, равной единице.

Разницу между оригинальным распределением и приведенным к ограниченному виду можно пронаблюдать на (рис. 4).

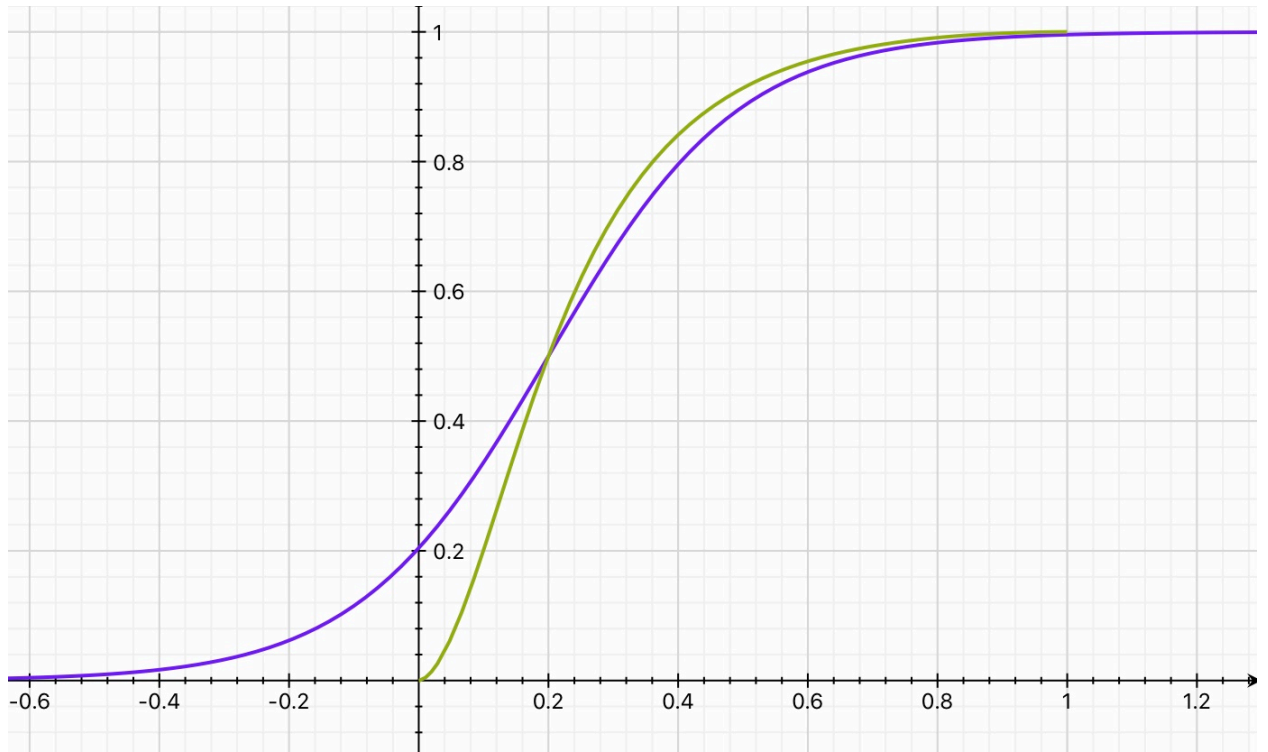


Рис. 4: Кривые ICC ( $c = 0$ ,  $b = 0.2$ )

Уровень знаний студента по каждой компетенции всегда хранится в виде вероятностного распределения в дискретном виде. При каждом верном или неверном ответе на тестовый вопрос это распределение изменяется в соответствии с формулой Байеса [12] для условной вероятности. Вопросы по возможности подбираются той сложности, что соответствует текущему уровню студента, для более эффективного обеспечения сходимости теоретической оценки к реальным показателям.

### 3.3.1. Адаптация сложности модулей

Сразу после прохождения модуля студенту предлагается выбрать степень понимания пройденного материала. В зависимости от ответа:

- определяется начальный уровень пройденных компетенций, который впоследствии может быть скорректирован прохождением тестовых заданий;
- может измениться сложность пройденного модуля, которая учитывается при создании индивидуальной траектории. Например, если материал не усвоен (что может также быть выяснено в результате тестирования), то при наличии альтернативы в следующий раз будет предложен другой модуль, обеспечивающий нужную компетенцию.

### 3.4. Организация данных

Для организации данных в приложении используется два различных серверных решения: Comapping<sup>3</sup> и Jsonbin<sup>4</sup>. Comapping используется для хранения основных материалов курса, а также относящихся к нему компетенций и тестовых заданий. В то время как Jsonbin используется для управления мета-информацией (данными, обеспечивающими адаптивность), включая информацию о студентах и процессе прохождения ими курса.

#### 3.4.1. Comapping API

MindMapping – это метод структуризации идей, мыслей и концепций графически в виде диаграммы (дерева) связей. Концепции или идеи связаны ветвями и отходят от некоторого центрального понятия (рис. 5).

Карты MindMap имеют множество областей применения, связанных со структуризацией различных понятий, включая конспектирование, brainstorming, планирование и проектирование проектов, и в частности обучение [5]. Их достоинством является скорость и легкость получения общего представления о заложенной информации за счёт визуализации.

---

<sup>3</sup><https://comapping.com/>

<sup>4</sup><https://jsonbin.io/>



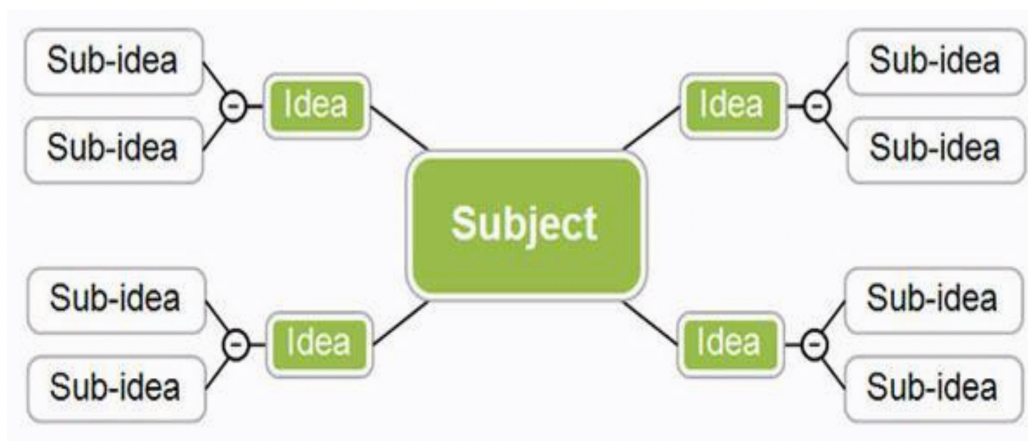


Рис. 5: MindMap

Comapping – это онлайн-система создания и редактирования карт MindMap, удобно представляющая информацию в виде масштабируемого дерева (рис. 6).

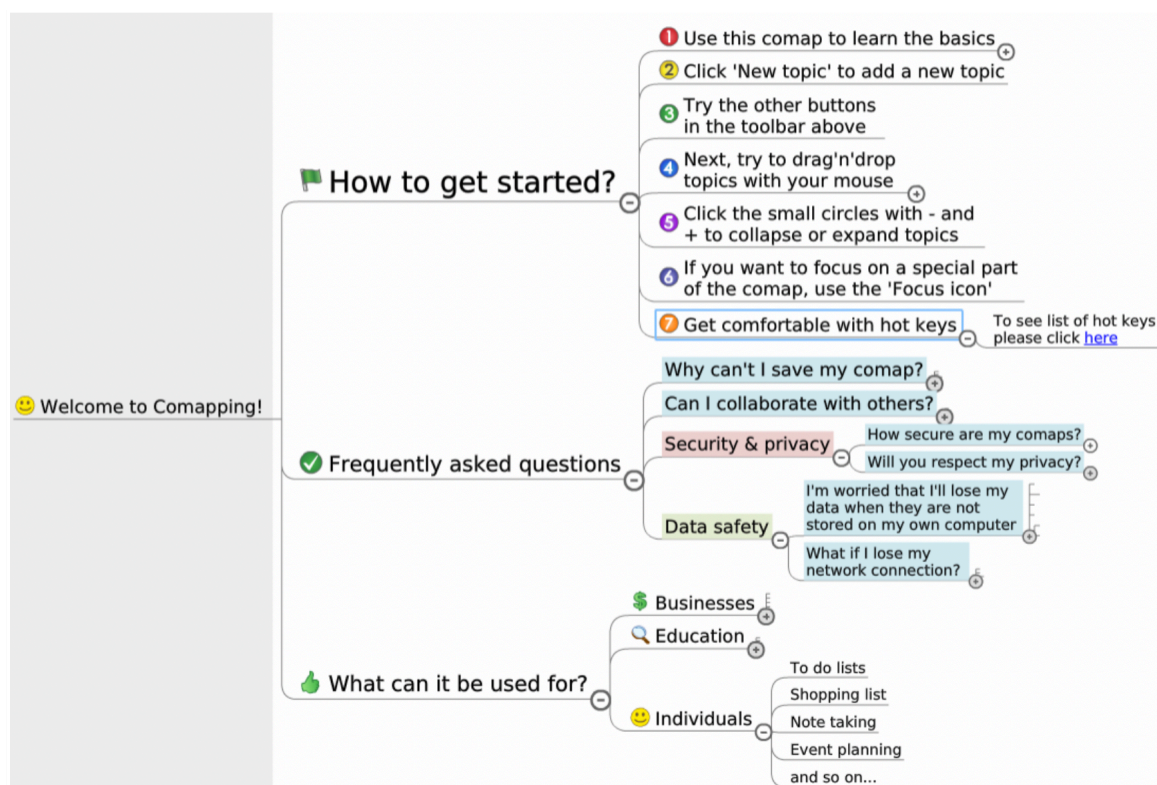


Рис. 6: Comapping

Comapping была выбрана в качестве основного инструмента для создания курсов в рамках приложения, предоставляя возможности для легкого проектирования структуры и хранения материалов в виде при-

крепленных к узлам файлов.

Материалы модулей курса могут храниться в различном виде, включая обычный текстовый формат, формат HTML или ссылку на существующую веб-страницу, и прикрепляются автором к узлам карты, построенной на Comapping. Что касается дерева компетенций, то оно строится в виде отдельной карты Comapping, в которой листьями помимо самих компетенций могут служить привязанные к ним тестовые задания, хранящиеся также в виде прикрепленных файлов.

Comapping предоставляет API [3], позволяющее взаимодействовать с сервером внутри приложения, и хранит карты пользователей в формате XML с интуитивно-понятной структурой, удобно описывающей дерево MindMap и его характеристики (рис. 7). Корневой элемент состоит из трех частей: мета-информации о самой карте (<metadata>), ее текущего визуального представления (<presentation>) и информации о содержимом. Узлы дерева хранятся внутри тега <node>, а их узлы-потомки представляются аналогичными вложенными тегами. Каждый узел сопровождается дополнительной информацией, такой как уникальный id, прикрепленный файл, иконка и прочее.

```
<node id="1058529" priority="1">
  <note><![CDATA[This is a note]]></note>
  <text><![CDATA[Use this comap to learn the basics]]></text>
  <node arrow="blue" flag="caution" id="1058530" smiley="happy"
star="yellow" taskCompletion="50">
    <icon name="clock"/>
    <task deadline="tomorrow" estimate="1 day" responsible="You"
start="today"/>
    <text><![CDATA[Don't worry - you can't break anything no
matter how hard you try!]]></text>
  </node>
</node>
```

Рис. 7: XML-представление карты

Для работы с XML документами в приложении используется библиотека SWXMLHash<sup>5</sup>.

<sup>5</sup><https://github.com/drmohundro/SWXMLHash>

### 3.4.2. Jsonbin API

Для всей остальной информации, хранящейся в приложении и необходимой для его работы, используется сервис Jsonbin, выбранный как простой способ серверного хранения данных для небольшого приложения.

Jsonbin позволяет хранить данные в JSON-формате и предоставляет REST API [7] со всеми необходимыми CRUD-операциями (запись, чтение, модификация и удаление). Для каждого курса и каждого студента на сервере Jsonbin выделяется отдельный JSON-документ, хранящий нужную мета-информацию.

Для работы с веб-запросами как в случае с Jsonbin, так и в случае с Comapping используется библиотека Alamofire<sup>6</sup>.

---

<sup>6</sup><https://github.com/Alamofire/Alamofire>

## 4. Эксперименты

В качестве эксперимента был создан адаптивный курс по спидкубингу, то есть сборке кубика Рубика на скорость. Курс был собран из готовых материалов, взятых из различных источников, и снабжен адаптивной разметкой и заданиями для тестирования. На рисунке 8 изображена полученная карта компетенций.



Рис. 8: Карта компетенций для курса – Comapping

Курс был предложен для прохождения двум людям с различной степенью начальной подготовки и осведомленности в теме: один из них был ознакомлен с предметом достаточно поверхностно, обладая однако некоторыми базовыми сведениями и навыками, в то время как второй занимается профессионально и участвует в соревнованиях по этому виду спорта.

Испытуемыми был выбран одинаковый набор целевых компетенций для возможности дальнейшего сравнения их результатов.

По итогам непрерывной корректировки индивидуальных показателей в процессе прохождения ими курса, тестируемые получили различные индивидуальные траектории, причем для одного из них (более

подготовленного) траектория получилась значительно короче, чем для второго.

Полученные результаты демонстрируют, что разработанная система действительно является адаптивной, поскольку подстраивается под каждого конкретного студента и в результате генерирует траекторию обучения, соответствующую его индивидуальным характеристикам.

## Заключение

В ходе работы было реализовано приложение, предоставляющее возможности для создания и прохождения курсов адаптивного обучения, и выполнены следующие задачи:

- Исследованы существующие разработки и алгоритмы в области адаптивного обучения;
- Разработана архитектура приложения;
- Подобраны и реализованы алгоритмы адаптивного обучения: поиска индивидуальной образовательной траектории и адаптивного тестирования;
- Реализована система адаптивного обучения с использованием выбранных алгоритмов;
- При помощи системы построен адаптивный курс и проанализированы результаты его прохождения реальными людьми.

## Список литературы

- [1] Atif Y., Benlamri Rachid, Berry Jawad. Learning Objects Based Framework for Self-Adaptive Learning. — 2003.
- [2] C. Lester James, Maria Vicari Rosa, Fábio Paraguacu. Intelligent Tutoring Systems. — 2004.
- [3] Comapping. Comapping API. — URL: <https://www.comapping.com/images/API/api%20reference.pdf> (дата обращения: 20.05.2020).
- [4] Competency-based Learning Object Sequencing using Particle Swarms / Luis de Marcos, Carmen Pagés, José Javier Martínez, José Antonio Gutiérrez. — 2008.
- [5] The Effect of Mind Mapping on Teaching and Learning: A Meta-Analysis / Ying Liu, Guoqing Zhao, Guozhen Ma, Yuwei Bo // Standard Journal of Education and Essay. — 2014. — 04. — Vol. 2. — P. 17–31.
- [6] Genetic algorithms for courseware engineering / Luis de Marcos, José Javier Martínez, José Antonio Gutiérrez et al. — 2011.
- [7] Jsonbin. Jsonbin API. — URL: <https://jsonbin.io/api-reference> (дата обращения: 20.05.2020).
- [8] Ree Malcolm James. Estimating Item Characteristic Curves // Applied psychological measurement. — 1979. — Vol. 3, no. 3. — P. 371–385.
- [9] Tadlaoui Mouenis Anouar, Khaldi Mohamed. Personalization and Collaboration in Adaptive E-Learning. — 2019.
- [10] Tecnologico de Monterrey. Edu Trends Adaptive Learning and Testing. — July 2014. — P. 21.
- [11] Zaporozhko Veronika V., Bolodurina Irina P., Parfenov Denis I. A genetic-algorithm approach for forming individual educational

trajectories for listeners of online courses // Proceedings of REMS 2018. — 2018.

- [12] van der Linden Wim J., Pashley Peter J. Item Selection and Ability Estimation in Adaptive Testing // Computerized Adaptive Testing: Theory and Practice. — 2000. — P. 1–25.
- [13] Кречетов И.А., Кручинин В.В. Об одном алгоритме адаптивного обучения на основе кривой забывания. — 2017.