



Выбор оптимального способа реализации инструментального средства управления обучением с помощью метода анализа иерархий

В.А. Латыпова

Уфимский государственный авиационный технический университет

Аннотация: Веб-разработка используется при реализации различных систем и средств в том случае, когда пользователи должны иметь возможность работать на различных компьютерах и дистанционно. К данным средствам относятся и инструментальные средства управления обучением, где основными пользователями являются студенты и преподаватели, которые участвуют в дистанционном, а также смешанном обучении. Существуют различные способы создания веб-приложений, обладающие как достоинствами, так и недостатками по ряду критериев. При выборе способа реализации веб-приложения для создания инструментального средства управления обучением возникает проблема многокритериального выбора оптимального решения. В статье определено три способа создания веб-приложений: с помощью компилируемых и компилируемо-интерпретируемых языков программирования, с помощью интерпретируемых скриптовых языков программирования и с помощью связки: системы управления контентом и конструктора контента. Выделены основные критерии способов создания веб-приложения, такие как: простота реализации, скорость реализации, опыт использования и поддержка. Для выбора оптимального способа реализации веб-приложения применен метод анализа иерархий. Использование данного метода позволило определить оптимальный способ реализации веб-приложения для создания инструментального средства управления обучением: с помощью скриптовых языков программирования.

Ключевые слова: многокритериальный выбор, метод анализа иерархий, способ создания веб-приложений, система управления контентом, конструктор контента, инструментальное средство управления обучением, дистанционное обучение, смешанное обучение.

Введение

Веб-разработка используется при реализации различных систем и средств в том случае, когда пользователи должны иметь возможность работать на различных компьютерах и дистанционно. К данным средствам относятся и инструментальные средства управления обучением, где основными пользователями являются студенты и преподаватели, участвующие в дистанционном и смешанном обучении.

Есть различные способы создания веб-приложений. Они обладают рядом как достоинств, так и недостатков, которые связаны со значением



основных характеристик существующих способов создания. Поэтому при выборе способа реализации веб-приложения для создания инструментального средства управления обучением возникает проблема многокритериального выбора оптимального решения. Для решения данной проблемы необходимо определить существующие способы создания веб-приложений, выделить основные критерии (наиболее важные характеристики) способов реализации и сравнить существующие способы по данным критериям для выбора оптимального решения.

1 Инструментальное средство управления обучением как необходимое дополнение к существующим системам дистанционного обучения

Для управления обучением используются различные системы дистанционного обучения (далее СДО). Они обладают однотипным функционалом и позволяют, например, следующее: проходить авторизацию, публиковать и читать учебные материалы (лекции, методические пособия), проходить тестирование и создавать тесты, загружать и скачивать работы на проверку. Но есть такие задачи, которые существующие системы не решают и необходимы внешние инструментальные средства. Например, при обучении студентов курсовому проектированию функции существующих систем сводятся к загрузке работы студентом, скачиванию и проставлению оценки за данную работу преподавателем. Такие работы относятся к классу сложных открытых задач (далее СОЗ) [1]. Стандартная проверка с помощью тестов для таких задач не приемлема. Поэтому для автоматизации проверки таких работ, а также эффективного управления обучением в данном случае функций существующих систем недостаточно. Вследствие этого возникает необходимость реализации инструментального средства, позволяющего управлять обучением при наличии СОЗ.

Инструментальное средство управления обучением позволяет собирать и обрабатывать статистические данные по процессу обучения, когда



студенты выполняют СОЗ. Рассчитываются такие показатели как: среднее количество попыток сдачи одной работы, среднее количество ошибок в работе, повторяемость ошибок у разных студентов (доля студентов, сделавших одну и ту же ошибку) и др.[2]

Инструментальное средство не имеет высоких требований по нагрузке. Предполагаются математические расчеты и запросы к базе данных, но т.к. пользователем является один преподаватель, то нагрузка не будет высокой. Программа должна разрабатываться как веб-приложение, чтобы обеспечить гибкую работу преподавателя, позволяя ему осуществлять смешанное и дистанционное обучение.

2 Существующие способы создания веб-приложений для реализации инструментального средства управления обучением

На данный момент существует три способа создания веб-приложений:

- с помощью компилируемых языков программирования (далее ЯП), таких как C, C++, Delphi и компилируемо-интерпретируемых – Java, C#;
- с помощью интерпретируемых скриптовых ЯП, таких как JavaScript, VBScript, Python, PHP и др.;
- с помощью систем управления контентом (Content management system – далее CMS), таких как Joomla, WordPress, Drupal, 1С-Битрикс.

При использовании компилируемых ЯП программа перед использованием проходит процедуру компиляции: код программы преобразуется в инструкции процессора, который будет ее исполнять. При использовании компилируемо-интерпретируемых ЯП программа компилируется в байт-код. Для исполнения такой программы нужна виртуальная машина, которая интерпретирует байт-код. Программы, написанные на компилируемых и компилируемо-интерпретируемых ЯП,



работают быстрее программ, созданных другим способом и позволяют выдерживать большую нагрузку.

При использовании интерпретируемых ЯП, программа в исходном виде исполняется интерпретатором. Скорость выполнения таких программ ниже.

При использовании CMS последняя выступает как ядро разрабатываемой системы. Здесь не требуется программирования как такового. Устанавливается стандартная CMS, а далее, по необходимости, расширяется установкой модулей и плагинов, которые могут выполнять различные стандартные для сайтов функции: отображение новостей, организация форума и т.п. Приложение собирается как конструктор. Скорость выполнения программ, созданных на основе CMS самая низкая.

Обучение созданию приложений занимает очень короткий срок, также для большинства CMS очень развита поддержка, в том числе русскоязычная. Однако при создании инструментального средства управления обучением использование только CMS не подойдет, т.к. функционал требуется специфичный, и существующего стандартного, используемого при создании сайтов, не достаточно. Поэтому в данном случае необходимо вручную дорабатывать значительное количество кода данной системы. А это приведет в будущем к проблемам с безопасностью, т.к. для поддержания последней необходимо регулярно обновлять версию CMS. При обновлении все введенные в код временные решения, дающие нужный функционал, будут удалены.

Решением данной проблемы является использование конструктора контента (Content Construction Kit – далее ССК). Данный конструктор позволяет без ручной доработки CMS в автоматизированном режиме создавать специфичные модули и плагины. В связке с CMS данный инструмент позволит реализовать необходимую функциональность. Но использование ССК несколько снижает скорость и простоту реализации, т.к.



необходимые функции разрабатываются самостоятельно, хоть и в автоматизированном режиме, а не просто устанавливается подходящее расширение. Также необходимы хотя бы небольшие знания по программированию. Трудностью также является лишь частичная поддержка: неполнота документации, и информация в основном на английском языке. Большинство ССК работают как отдельные приложения: у них нет полной интеграции с CMS, и они, соответственно, не работают с ней напрямую. Это приводит к снижению производительности. Очень небольшая часть ССК имеют полную интеграцию с CMS. Например, для такой популярной CMS как Joomla единственным ССК, работающим напрямую с ядром, является ССК Seblod [3].

Первый способ реализации для разработки самый трудоемкий по сравнению с остальными способами и требует много времени на создание приложения. Он ориентирован больше на реализацию ресурсоемких приложений, что в данном случае не требуется. Есть много литературы по такому типу программирования. Второй способ реализации проще первого. Но он сложнее третьего способа. Литературы также достаточно. Третий способ реализации самый простой и быстрый.

Все три способа создания веб-приложения имеют преимущества и недостатки, поэтому возникает проблема выбора способа реализации инструментального средства управления обучением.

3. Использование метода анализа иерархий при выборе способа создания инструментального средства управления обучением

Для решения проблемы выбора способа реализации инструментального средства управления обучением выбран метод анализа иерархий, разработанный Саати [4], [5]. Данный метод используется в различных областях, а применительно к разработке и выбору систем используется для выбора программного обеспечения [6], для выбора системы управления



базами данных [7], для выбора элементов пользовательского интерфейса программы [8], для выбора системы электронного документооборота [9], для выбора систем защиты информации от утечек [10]. Данный метод обладает рядом достоинств, которые описаны в работе [11]:

- попарное сравнение критериев и альтернатив (что является предпочтительным для эксперта из-за отсутствия необходимости оперировать сразу несколькими сравниваемыми объектами);
- возможность изменения состава критериев и альтернатив без полного пересмотра результатов опроса;
- использование вербально-числовой шкалы, что позволяет производить сравнение критериев, которые могут быть выражены как количественно, так и качественно.

Объектом в проблеме является способ реализации. Данный объект характеризуется рядом основных, выделенных автором, критериев:

1. простота реализации;
2. скорость реализации;
3. опыт использования;
4. поддержка.

Критерий «Простота реализации» показывает, сколько усилий необходимо для реализации средства. Критерий «Скорость реализации» показывает, сколько времени необходимо для реализации системы. Критерий «Опыт использования» показывает уровень знания и опыта работы со средой программирования программиста. Критерий «Поддержка» показывает наличие литературы, форумов и сообществ, облегчающих реализацию системы. Критерий «Скорость исполнения программы» не выделен в виду того, что программа не будет ресурсоемкой, и поэтому все три способа по данному критерию будут равноправны.

Альтернативами решения проблемы являются, соответственно, приведенные выше 3 способа реализации: с помощью компилируемых и компилируемо-интерпретируемых ЯП, с помощью интерпретируемых скриптовых ЯП, с помощью связки: CMS и ССК. Иерархия целей представлена на рис.1.

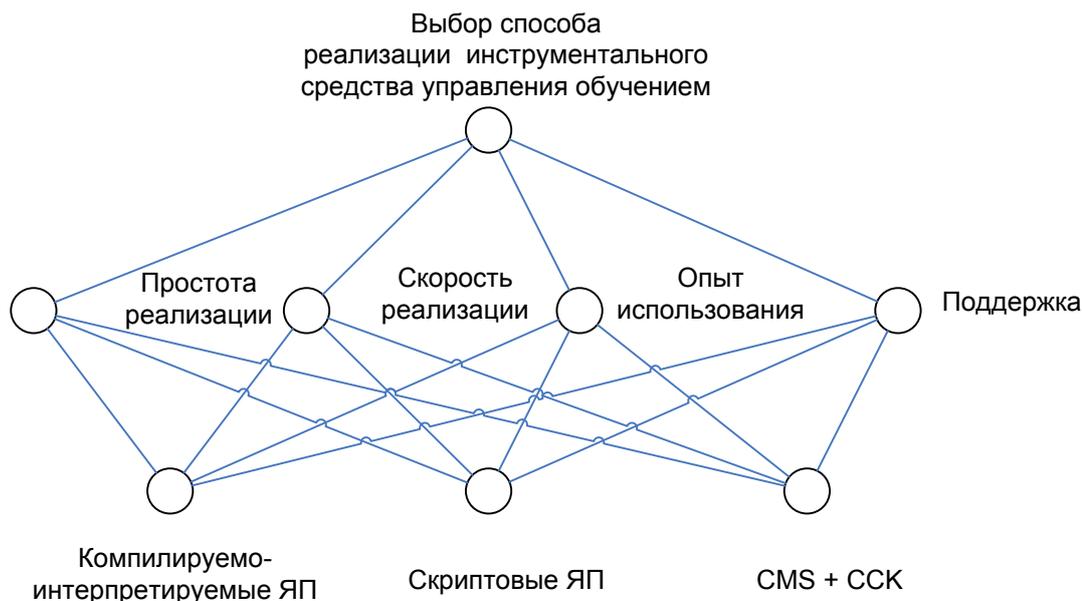


Рис.1. – Иерархия целей при выборе способа реализации инструментального средства управления обучением

В таблице №1 представлены качественные значения критериев для альтернатив, полученные экспертно.

Таблица №1

Качественные значения критериев для альтернатив

Альтернатива	Критерий			
	Простота реализации	Скорость реализации	Опыт использования	Поддержка
Компилируемо-интерпретируемые ЯП	низкая	низкая	небольшой	полная



Скриптовые ЯП	средняя	низкая	большой	полная
CMS +ССК	выше среднего	средняя	ниже среднего	неполная

Для третьей альтернативы указана простота реализации выше среднего, а не высокая, из-за необходимости работы с ССК. По этой же причине указана неполная поддержка.

Шкала попарного сравнения, используемая в методе анализа иерархий представлена в таблице №2.

Таблица №2

Шкала попарного сравнения [4]

Степень значимости	Определение
1	Одинаковая значимость
3	Слабая значимость
5	Сильная значимость
7	Очень сильная значимость
9	Абсолютная значимость
2, 4, 6, 8	Промежуточное значение

Сначала попарно сравниваются критерии, и строится соответствующая матрица суждений. Далее определяется вектор приоритетов критериев. Для получения вектора приоритетов вычисляется главный собственный вектор матрицы суждений и производится его нормализация. Приблизительно вычислить вектор приоритетов с высокой точностью можно умножением элементов строки матрицы с извлечением корня степени, равной рангу матрицы [4]. Компоненты главного собственного вектора V матрицы суждений R вычисляются по формуле:

$$v_i = \sqrt[k]{\prod_{j=1}^k r_{ij}}$$



где k – ранг матрицы суждений; i – номер компонента собственного вектора, $i = 1, 2 \dots k$.

Нормализация вектора V , для вычисления компонент вектора приоритетов V_n выполняется по формуле:

$$v_{in} = \frac{v_i}{\sum_{i=1}^k v_i}$$

В таблице №3 представлена матрица попарного сравнения критериев с компонентами главного собственного вектора и вектора приоритетов.

Таблица №3

Матрица попарного сравнения критериев

Критерий	Простота реализации	Скорость реализации	Опыт использования	Поддержка	Компонента собственного вектора	Компонента вектора приор.
Простота реализации	1	6	5	3	3,0801	0,5531
Скорость реализации	$\frac{1}{6}$	1	$\frac{1}{3}$	$\frac{1}{5}$	0,3247	0,0583
Опыт использования	$\frac{1}{5}$	3	1	$\frac{1}{3}$	0,6687	0,1201
Поддержка	$\frac{1}{3}$	5	3	1	1,4953	0,2685
Сумма	1,7	15	9,3333	4,5333	5,5688	1

Сравнение критериев проводится по шкале, представленной выше.

Для оценки согласованности мнений экспертов рассчитывается главное (максимальное) собственное значение матрицы суждений L_{max} . Простым способом его получения является следующий способ: суммирование элементов матрицы попарного сравнения по столбцам и умножение полученного вектора на вектор приоритетов [5]:

$$L_{max} = S * V_n,$$



где S – вектор сумм столбцов матрицы суждений.

Индекс согласованности, показывающий отклонение от согласованности в суждениях определяется по формуле [4]:

$$ИС = \frac{L_{\max} - k}{k - 1},$$

Отношение согласованности $ОС$ рассчитывается по формуле [4]:

$$ОС = \frac{ИС}{СИ},$$

где $СИ$ – средний случайный индекс, значение которого зависит от разрядности матрицы и представлено в работе [4].

Условие, чтобы согласованность была приемлема, имеет вид:

$$ОС \leq 0,1.$$

Для представленной выше матрицы суждений получены следующие значения: $L_{\max} = 4,1529$; $ИС = 0,051$; $СИ = 0,9$; $ОС = 0,0567$.

Следовательно, мнения эксперта согласованы.

После сравнения критериев проводится попарное сравнение альтернатив по каждому критерию. В таблице №4 представлена матрица попарного сравнения альтернатив по критерию «Простота реализации».

Таблица №4

Матрица сравнения альтернатив по критерию 1 «Простота реализации»

Критерий	Компилируемо-интерпретируемые ЯП	Скриптовые ЯП	CMS + ССК	Компонента собственного вектора	Компонента вектора приоритетов
Компилируемо-интерпретируемые ЯП	1	$\frac{1}{5}$	$\frac{1}{7}$	0,3057	0,0751



Скриптовые ЯП	5	1	$\frac{1}{2}$	1,3572	0,3332
CMS + ССК	7	2	1	2,4099	0,5917
Сумма	13	3,2	1,6429	4,0728	1

$L_{\max} = 3,0146$; $ИС = 0,0073$; $СИ = 0,58$; $ОС = 0,0126$. Мнения эксперта согласованы.

В таблице №5 представлена матрица попарного сравнения альтернатив по критерию «Скорость реализации».

Таблица №5

Матрица сравнения альтернатив по критерию 2 «Скорость реализации»

Критерий	Компилируемо-интерпретируемые ЯП	Скриптовые ЯП	CMS + ССК	Компонента собственного вектора	Компонента вектора приоритетов
Компилируемо-интерпретируемые ЯП	1	1	$\frac{1}{5}$	0,5848	0,1429
Скриптовые ЯП	1	1	$\frac{1}{5}$	0,5848	0,1429
CMS + ССК	5	5	1	2,9237	0,7143
Сумма	7	7	1,4	4,0933	1

$L_{\max} = 3,0006$; $ИС = 0,0003$; $ОС = 0,0005$. Мнения эксперта согласованы.

В таблице №6 представлена матрица попарного сравнения альтернатив по критерию «Опыт использования».

Таблица №6

Матрица сравнения альтернатив по критерию 3 «Опыт использования»



Критерий	Компилируемо- интерпретируемые ЯП	Скриптовые ЯП	CMS + ССК	Компонента собственного вектора	Компонента вектора приоритетов
Компилируемо- интерпретируемые ЯП	1	$\frac{1}{9}$	$\frac{1}{3}$	0,3334	0,0658
Скриптовые ЯП	9	1	7	3,9785	0,7853
CMS +ССК	3	$\frac{1}{7}$	1	0,754	0,1488
Сумма	13	1,254	8,3333	5,0659	1

$L_{\max} = 3,0798$; $ИС = 0,0399$; $ОС = 0,0688$. Мнения эксперта согласованы.

В таблице №7 представлена матрица попарного сравнения альтернатив по критерию «Поддержка».

Таблица №7

Матрица сравнения альтернатив по критерию 4 «Поддержка»

Критерий	Компилируемо- интерпретируемые ЯП	Скриптовые ЯП	CMS + ССК	Компонент собственного вектора	Компонент вектора приоритетов
Компилируемо- интерпретируемые ЯП	1	1	7	1,9128	0,4667
Скриптовые ЯП	1	1	7	1,9128	0,4667
CMS +ССК	$\frac{1}{7}$	$\frac{1}{7}$	1	0,2733	0,0667



Сумма	2,1429	2,1429	15	4,0989	1
-------	--------	--------	----	--------	---

$L_{\max} = 3,0007$; $ИС = 0,0004$; $ОС = 0,0007$. Мнения эксперта согласованы.

Выбор оптимального решения производится исходя из глобальных приоритетов альтернатив. Вектор глобальных приоритетов рассчитывается как произведение матрицы, состоящей из векторов приоритетов альтернатив, и вектора приоритетов критериев [4]. Матрица из векторов приоритетов альтернатив представлена в таблице №8.

Таблица №8

Матрица, состоящая из векторов приоритетов альтернатив

Альтернатива	Простога реализации	Скорость реализации	Опыт использования	Поддержка
Компилируемо-интерпретируемые ЯП	0,0751	0,1429	0,0658	0,4667
Скриптовые ЯП	0,3332	0,1429	0,7853	0,4667
CMS +ССК	0,5917	0,7143	0,1488	0,0667

Вектор глобальных приоритетов альтернатив отображен в таблице №9.

Таблица №9

Вектор глобальных приоритетов

Альтернатива	Глобальный приоритет
Компилируемо-интерпретируемые ЯП	0,1831



Скриптовые ЯП	0,4122
CMS +ССК	0,4047

Максимальный глобальный приоритет у альтернативы «Скриптовые ЯП», поэтому для создания инструментального средства управления обучением оптимальным является этот способ реализации.

Выводы

Определены три способа создания веб-приложений: с помощью компилируемых и компилируемо-интерпретируемых ЯП, с помощью интерпретируемых скриптовых ЯП, с помощью связки CMS и ССК. Выделены основные критерии способов создания, такие как: простота реализации, скорость реализации, опыт использования и поддержка. Для выбора оптимального способа применен метод анализа иерархий. Использование данного метода позволило определить способ реализации веб-приложения с помощью скриптовых ЯП для создания инструментального средства управления обучением при наличии СОЗ.

Литература

1. Латыпова В.А. Сложные открытые задачи в смешанном и дистанционном автоматизированном обучении // Инженерный вестник Дона. 2015. №3. URL: ivdon.ru/ru/magazine/archive/n3y2015/3211.
2. Латыпова В.А. Оценка эффективности процесса обучения при наличии сложных открытых задач с помощью экспертных методов // Инженерный вестник Дона. 2016. №1. URL: ivdon.ru/ru/magazine/archive/n1y2016/3540.
3. Seblod URL: seblod.com (Free access).
4. Саати Т. Принятие решений/Метод анализа иерархий. М.: Радио и связь, 1993. 278 с.



5. Saaty T.L., Vargas L.G. Models, methods, concepts & applications of the analytic hierarchy process. New York: Springer Science+Business Media, 2001. 333 p.

6. Гагарин А.Г. Анализ качества программного обеспечения методом анализа иерархий: проблема учета влияния находящихся на границах области допустимых значений оценок при большом количестве критериев и альтернатив // Аудит и финансовый анализ. 2010. №2. URL: auditfin.com/fin/2010/2/11_01.pdf.

7. Земцов А. Н., Болгов Н. В., Божко С. Н. Многокритериальный выбор оптимальной системы управления базы данных с помощью метода анализа иерархий // Инженерный вестник Дона. 2014. №2. URL: ivdon.ru/magazine/archive/n2y2014/2360.

8. Полевщиков И. С. Методика выбора элементов пользовательского интерфейса программы с применением метода анализа иерархий (часть 1) // Молодой ученый. 2016. №3. С. 59-61.

9. Глухова Ю.В. Использование метода анализа иерархий для выбора системы электронного документооборота // Современные научные исследования и инновации. 2014. № 4. URL: web.snauka.ru/issues/2014/04/33612

10. Зверев И. Н. Применение метода анализа иерархий при сравнении DLP-систем // Естественные и математические науки в современном мире: сб. ст. по матер. XX междунар. науч.-практ. конф. № 7(19). Новосибирск: СибАК, 2014. URL: sibac.info/conf/naturscience/xx/38755.

11. Коробор В.Б., Тутьгин А.Г. Преимущества и недостатки метода анализа иерархий // Известия РГПУ им. А.И. Герцена. 2010. №122. С.108-115.

References

1. Latypova V.A. Inženernyj vestnik Dona (Rus). 2015. №3. URL: ivdon.ru/ru/magazine/archive/n3y2015/3211.



2. Latypova V.A. Inzhenernyj vestnik Dona (Rus). 2016. №1. URL: ivdon.ru/ru/magazine/archive/n1y2016/3540.
3. Seblod URL: seblod.com (Free access).
4. Saati T. Prinyatie resheniy/Metod analiza ierarkhiy [Decision making / Analytic Hierarchy Process]. M.: Radio i svyaz', 1993. 278 p.
5. Saaty T.L., Vargas L.G. Models, methods, concepts & applications of the analytic hierarchy process. New York: Springer Science+Business Media, 2001. 333 p.
6. Gagarin A.G. Audit i finansovyy analiz. 2010. №2. URL: auditfin.com/fin/2010/2/11_01.pdf.
7. Zemtsov A. N., Bolgov N. V., Bozhko S. N. Inzhenernyj vestnik Dona (Rus). 2014. №2. URL: ivdon.ru/magazine/archive/n2y2014/2360.
8. Polevshchikov I. S. Molodoy uchenyy. 2016. №3. S. 59-61.
9. Glukhova Yu.V. Sovremennye nauchnye issledovaniya i innovatsii. 2014. № 4. URL: web.snauka.ru/issues/2014/04/33612.
10. Zverev I. N. Estestvennye i matematicheskie nauki v sovremennom mire: sb. st. po mater. XX mezhdunar. nauch.-prakt. konf. № 7(19). Novosibirsk: SibAK, 2014. URL: sibac.info/conf/naturscience/xx/38755.
11. Korobor V.B., Tutygin A.G. Izvestiya RGPU im. A.I. Gertsena. 2010. №122. S.108-115.