

References

1. Visintin A. (1993) Hysteresis and semigroups, in “Models of Hysteresis”// A.Visintin, ed. Longman, Harlow. P.192-206.
2. M.Hilpert M. (1989) On uniqueness for evolution problems with hysteresis // In: Mathematical Models for Phase Change Problems. Birkhauser,Basel. P. 377-388.
3. Aliev A.B., Isayeva S.E. (2015) A global attractor for one semilinear hyperbolic equation with memory operator, Pleiades Publishing Ltd, Computational Mathematics and Mathematical Physics. vol 55. №11.
4. Krejci P. (1986) Hysteresis and periodic solutions of semilinear and quasilinear wave equations// Math.Z. 193. P. 247-264.
5. Krejci P. (1993) Asymptotic stability of periodic solutions to the wave equation with hysteresis// In: Models of hysteresis (A.Visintin, ed.). Longman, Harlow. P. 77-90.
6. Visintin A. (1993) Differential Models of Hysteresis. Springer. 411 p.
7. Leeons Z.L. (1972) Nekotory`e metody` resheniia nelinei`ny`kh kraevy`kh zadach [Some methods for solving nonlinear boundary value problems] M.: Mir, 1972.

УДК
004.9

**ИНТЕРАКТИВНЫЙ МОТИВАТОР
ИЗУЧЕНИЯ ПРЕДМЕТА «ФЭМУЛАС»**

Светлана Юрьевна Петрова | Новгородский государственный
к.т.н., доцент | университет им. Ярослава Мудрого
svetlana.petrova@novsu.ru
г. Великий Новгород

Аннотация. В статье рассматривается вопрос использования информационной технологии Фэмулас в образовании. Современная система образования сталкивается с проблемами совершенствования технологий самостоятельной работы студентов и организации инклюзивного образования инвалидов, особенно если студент слабо мотивирован в обучении. Выяснилось, что большинство студентов не в состоянии определить ценность публикации и самостоятельно найти нужные знания по проблемному предмету. Таким образом была поставлена задача разработки интерактивного мотиватора, позволяющего максимально упростить подборку актуальной, интересной и качественной информации по изучаемому предмету. Сбор данных для мотиватора осуществляет бот, основная функция которого направлена на обнаружение событий взаимодействия эксперта в изучаемой студеном предметной области с цифровыми образовательными ресурсами портала НовГУ и Интернет. Полученные данные анализируются с помощью технологического решения Фэмулас, формирующего глобальную историю причинно-следственных связей взаимодействия эксперта с цифровыми образовательными ресурсами и определения важности страницы. Во время фазы спецификации причинно-следственных связей переход пользователя по цифровым ресурсам мы рассматриваем как конечный автомат и определяем действия, которые вызывают переходы от одного цифрового ресурса к другому. Причинно-следственная связь локальных событий может быть получена из истории процесса. Два события в глобальной истории могут быть связаны. Если это так, ни одно из них не является причиной другого, следовательно, можно сказать, что такие события – это параллельные события. Полученные истории проходят процедуру обработки больших данных на вычислительном кластере Nadoor в НовГУ. Результатом обработки будет список ссылок на http страницы с актуальными, интересными и качественными публикациями по изучаемому предмету.

Ключевые слова: информационной технологии образования, большие данные, экспертные системы, персонализация образования.

Мировое образовательное пространство изменилось, на смену старым решениям приходят новые, например, смешанное обучение (Blended Learning). Мобильность – основной девиз современности. В связи с этим меняется вся инфраструктура информационного взаимодействия педагога со своими учениками, и во главу ставятся такие принципы как персонализация и мобильность образовательного процесса. Современная система образования сталкивается с проблемами совершенствования технологий самостоятельной работы студентов и организации инклюзивного образования инвалидов, особенно если студент слабо мотивирован в обучении. Изучая мировой опыт смешанного обучения, мы обнаружили, что одним из ключевых факторов успеха является привлечение внимания студента интересными публикациями в Интернете. Но вот в чем проблема. Как выяснилось, большинство из студентов не в состоянии определить ценность публикации и самостоятельно найти нужные знания по предмету. Интернет заполнен огромным количеством данных, качество которых сомнительно. Таким образом была поставлена задача разработки интерактивного мотиватора, позволяющего максимально упростить подборку актуальной, интересной и качественной информации по изучаемому предмету.

Технологическое решение заключается в реализации следующего программного обеспечения: 1) разработка бота для сбора данных; 2) разработка морфологического анализатора для экспертизы и сортировки контента (фильтрация данных); 3) разработка облачного сервиса Фэмулас для интеграции данных и предоставления интерактивного взаимодействия мотиватора с этими данными; 4) разработка рекомендательного Виджета для мобильного и настольного решения.

Первый этап – сбор данных. Собрать данные можно различными способами, которые категоризируются по типу участия: непосредственное общение или опосредованное, и методу сбора данных: прямой, косвенный, наблюдение и метод само-обследования. Использование нескольких методов, помогают открыть различные аспекты деятельности пользователя, и дают уверенность, что найдена значимая информация, а не просто получен странный результат от одного метода.

Одной из частей технологии Фэмулас является бот, деятельность которого направлена на обнаружение событий взаимодействия эксперта в предметной области с цифровой информацией. Бот представляет собой программу (например, приложение для мобильного устройства), исполняемую в клиентской системе. К событиям взаимодействия эксперта с цифровой информацией можно отнести события, вызванные взаимодействием эксперта с информационным порталом НовГУ или другим образовательным цифровым ресурсом Интернет. Бот собирает данные, связанные с зарегистрированными событиями взаимодействия студента, и отправляет собранные данные на сервер. В некоторых случаях бот отправляет HTTP-запрос на сервер, в котором собранные данные кодируются в параметрах HTTP-запроса.

Процесс обнаружения одного или нескольких событий взаимодействия эксперта с цифровым ресурсом выглядит следующим образом. Событие создается человеческим взаимодействием с приложением, запущенным на клиентской системе. Пользователь обычно взаимодействует с клиентской системой с помощью стандартных устройств ввода/вывода, таких как мышь или клавиатура, или сенсорных экранов и устройств распознавания речи. На первом этапе выполняется инициализация бота. Инициализация включает в себя загрузку бота в клиентскую систему с сервера и его

активацию. Бот – это программа, которой доверяет сервер. Программа может быть написана на различных языках программирования, таких как Java, JavaScript или Python [Bot Code Examples, 2017]. В некоторых вариантах реализации бот может быть загружен с сервера и работать в контексте пользовательского агента, такого как клиент браузера. Например, бот может содержать JavaScript-код, который отслеживает и регистрирует деятельность пользователя. Или бот может быть Java-программой, которая работает на виртуальной машине Java, которая сама по себе может быть плагином для пользовательского агента, такого как клиент браузера. Инициализация может быть инициирована событием передачи фокуса управления, генерируемые щелчками мыши или нажатием клавиш на клавиатуре.

В клиентской системе может быть несколько приложений. Боты могут работать со всеми приложениями, отслеживая взаимодействия и регистрируя события информационного характера. Инициализация бота обычно запускается, когда пользователь перемещает фокус управления (например, указатель мыши) на одно из запущенных в клиентской системе приложений. Как правило, бот инициализируется в каждом запущенном приложении в клиентской системе. Например, в клиентской системе может быть несколько приложений интернет-браузера. Бот может быть программой JavaScript, встроенной в веб-страницу, и инициализироваться, когда приложение браузера получает фокус управления.

После инициализации бот проверяет, были ли обнаружены события взаимодействия человека с цифровыми образовательными ресурсами. В частности, регистрация метаданных цифровых ресурсов, с которыми осуществлялись манипуляции, и накопление данных в буфере, который затем обрабатывается операционной системой. Бот по сути подписывается на события ввода/вывода передаваемых через браузер. Бот периодически просыпается и ждет обнаружения подписанных событий, таких как событие уведомления о перемещении фокуса управления. Если какие-либо подписанные события были переданы боту во время пробуждения, он собирает данные о событии и метаданные о цифровом ресурсе, с которым взаимодействовал пользователь. Собранные данные отправляются на сервер в хранилище в виде файла данных, внедренного в http-запрос.

Валидация и анализ данных могут проводиться как в реальном времени, то есть, как только данные поступят на сервер, так и в автономном режиме, то есть при анализе журнала регистрации событий взаимодействия. В некоторых случаях сервер может корректировать политику регистрации событий. Например, сервер может игнорировать клики на рекламных объявлениях.

Процесс регистрации запросов в журнале сервера, генерируемых ботами, выглядит следующим образом. На первом этапе процесса бот получает идентификатор. Бот запускается в клиентской системе и взаимодействует с сервером. В журнале регистрации запросов сохраняются записи с одного или нескольких ботов в одной или нескольких клиентских системах. Когда бот взаимодействует с сервером, он отправляет http-запрос серверу для получения файла с предопределенным именем. Когда сервер получает http-запрос, он создает запись в журнале, указывающую на предопределенное имя файла и другие параметры в http-запросе. Для обеспечения безопасной передачи данных могут использоваться протоколы шифрования. Декодирование записи в журнале, позволяет получить IP-адрес, идентификатор бота, http-файл cookie и данные, связанные с событиями взаимодействия студента с информационным ресурсом портала НовГУ. Http-файлы cookie используются веб-серверами для идентификации пользователей и для поддержания данных, связанных с одним и тем же сеансом пользователя во время нескольких http-запросов.

Полученные данные анализируются и формируется глобальная история причинно-следственных связей взаимодействия эксперта с цифровыми образовательными ресурсами, а также определяется важность страницы. Обработка поступивших данных, для формирования списка, происходит с помощью технологического решения Фэмулас, которое задействует облачные вычисления.

Для ранжирования страниц требуется четкое понимание причинно-следственных связей поступивших данных. Во время фазы спецификации траектории, переход пользователя по цифровым ресурсам, мы рассматриваем как конечный автомат и определяем действия, которые вызывают переходы от одного цифрового ресурса к другому. Во время фазы анализа журнала данных, полученных от бота, мы должны определить причину, которая привела эксперта к определенному образовательному или учебно-значимому цифровому ресурсу.

Деятельность любого процесса поиска образовательного цифрового ресурса моделируется как последовательность событий, следовательно, бинарное отношение причинно-следственной связи должно быть выражено в терминах событий и должно выражать наше знание о том, что причина должна предшествовать следствию. Так же, мы должны различать локальные и коммуникационных события.

События, влияющие более чем на один процесс, являются значимыми для построения глобальной истории нескольких процессов [Maginescu, 2013]. Обозначим h_i как локальную историю процесса p_i и пусть e_i^k обозначает i -ое событие в этой истории.

Двоичная причинно-следственная зависимость между двумя событиями, обладает следующими свойствами:

Причинно-следственная связь локальных событий может быть получена из истории процесса:

$$\text{если } e_i^k, e_i^l \in h_i \text{ и } k < l \text{ тогда } e_i^k \rightarrow e_i^l. \quad (1)$$

Причинно-следственная связь коммуникационных событий:

$$\text{если } e_i^k = \mathcal{S}(m) \text{ и } e_j^l = \mathcal{R}(m) \text{ тогда } e_i^k \rightarrow e_j^l. \quad (2)$$

Транзитивность причинно-следственной связи:

$$\text{если } e_i^k \rightarrow e_j^l \text{ и } e_j^l \rightarrow e_m^n \text{ тогда } e_i^k \rightarrow e_m^n. \quad (3)$$

Два события в глобальной истории могут быть связаны. Если это так, ни одно из них не является причиной другого, следовательно, можно сказать, что такие события, это параллельные события. Логические часы \mathcal{LC} являются необходимой абстракцией для обеспечения условия синхронизации в отсутствии глобальных часов. Каждый процесс p_i отображает события в целых положительных числах. Назовем $\mathcal{LC}(e)$ значением локальной переменной, связанной с событием e . Каждая временная метка процесса для каждого отправленного сообщения m со значением логических часов на время отправки $\mathcal{TS}(m) = \mathcal{LC}(\mathcal{S}(m))$. Правила для обновления логических часов задаются следующим соотношением:

$$\mathcal{LC}(e) = \begin{cases} \mathcal{LC} + 1 & \text{если } e \text{ это локальное событие, или } \mathcal{S}(m) \\ \max(\mathcal{LC} + \mathcal{TS}(m) + 1) & \text{если } e = \mathcal{R}(m) \end{cases} . \quad (4)$$

Процесс характеризуется его состоянием; состояние является совокупностью информации, которая нам нужна. Событие является изменением состояния процесса.

События, влияющие на состояние процесса p_1 нумеруются последовательно, как $e_1^1, e_1^2, e_1^3, \dots$. Каждый процесс последовательно маркирует свои локальные события и события отправки сообщений, пока не получит сообщение, отмеченное значением логических часов, которое больше, чем следующее значение локальных логических часов, как показано в уравнении (4). Отсюда следует, что логические часы не позволяют формировать глобальное упорядочение всех событий.

Предположим есть три процесса и набор событий с маркировкой $e_1^1, e_1^2, e_1^3, \dots$. Процесс p_1 находится в состоянии σ_i^j сразу после возникновения события e_i^j и остается в этом состоянии до наступления следующего события e_i^{j+1} . Все события в процессе p_1 локальные; процесс в состоянии σ_1 сразу после наступления события e_1^1 и остается в этом состоянии до наступления события e_1^2 . Два процесса p_1 и p_2 ; Событие e_2^1 является коммуникационным событием, p_1 отправляет сообщение p_2 ; e_2^3 является коммуникационным событием, процесс p_2 получает сообщение, посланное p_1 . Три процесса взаимодействуют посредством коммуникационных событий.

Корреляция между событиями и значениями логических часов следующая: $e_1^1, e_1^1, e_3^1 \rightarrow 1, e_1^5 \rightarrow 5, e_2^4 \rightarrow 7, e_3^4 \rightarrow 10, e_1^6 \rightarrow 12$ и так далее.

Глобальное упорядочение всех событий не представляется возможным, так как отсутствует возможность установить порядок событий e_1^1, e_1^2 и e_1^3 . Тем не менее, коммуникационные события позволяют этим трем процессам согласовывать свои логические часы. Например, процесс p_2 маркирует событие e_2^3 под номером 6, так как сообщение m_2 , которое несет информацию о значении логических часов было под номером 5, в то же самое время было отправлено сообщение m_3 . Напомним, что e_i^j это - ое событие в процессе p_i .

Логические часы нуждаются в важном свойстве – обнаружение разрыва. Рассмотрим два события e и e' , значения их логических часов $LC(e)$ и $LC(e')$, невозможно установить, если существует событие e'' такое что:

$$LC(e) < LC(e'') < LC(e'). \quad (5)$$

Например, для процесса p_1 есть событие e_1^4 расположенное между событиями e_1^3 и e_1^5 , поэтому $LC(e_1^3) = 3$, $LC(e_1^5) = 5$, $LC(e_1^4) = 4$ и $LC(e_1^3) < LC(e_1^4) < LC(e_1^5)$. Однако для процесса p_3 событие e_3^3 и e_3^4 следуют как $LC(e_3^3) = 3$ и $LC(e_3^4) = 10$.

В экспертной системе требуется знание о причинно-следственных связях всех процессов, чтобы сформировать глобальное состояние. Бот отвечает за построение глобального состояния, он отправляет информацию о локальных состояниях каждого процесса. Построение глобального состояния эквивалентно созданию моментальных снимков отдельных процессов информационного взаимодействия студента с цифровым образовательным ресурсом, а экспертная систем объединяет эти снимки в глобальное представление о состоянии системы. Объединение моментальных снимков — это несложный процесс, если все процессы имеют доступ к глобальным часам и снимки сделаны в одно и то же время и моментальные снимки будут соответствовать друг другу.

Причинно-следственная связь R – полное упорядочение всех событий в глобальной истории информационного взаимодействия в соответствии с локальной историей каждого процесса, участвующего в вычислениях:

$$R = (e_1^{j_1}, e_2^{j_2}, \dots, e_n^{j_n}). \quad (6)$$

Можно построить трехмерную решетку глобальных состояний начиная с инициации состояния $\Sigma^{(000)}$ и переходя в другое достижимое состояние $\Sigma^{(ijk)}$ с событиями i, j, k для процессов p_1, p_2, p_3 соответственно.

$R_1 = e_1^1, e_2^1, e_3^1, e_1^2$ соответствует как локальной истории каждого процесса, так и глобальной истории. Она является валидной, так как система прошла глобальные состояния:

$$\Sigma^{(000)}, \Sigma^{(100)}, \Sigma^{(110)}, \Sigma^{(000)}, \Sigma^{(111)}, \Sigma^{(211)}. \quad (7)$$

С другой стороны $R_2 = e_1^1, e_1^2, e_3^1, e_1^3, e_3^2$ является не валидным, потому что он противоречит глобальной истории. Система никогда не может достичь состояния $\Sigma^{(301)}$.

Сечение представляет собой подмножество локальных историй всех процессов. Если h_i^j обозначает историю процесса p_i включая его j -ое событие e_i^j , то сечение C является n -кортежем

$$C = \{h_i^j\} \text{ с } i \in \{1, n\} \text{ и } j \in \{1, n_i\}. \quad (8)$$

Граница сечения – это n -кортеж, состоящий из последнего события каждого процесса, включенного в сечение. Сечения обеспечивают необходимые знания для генерации глобального состояния, на основе обмена сообщениями между ботом и группой процессов. Для группы из трех процессов p_1, p_2, p_3 , описанных выше, можно сформировать два сечения, C_1 и C_2 . C_1 имеет границу основанную на событиях (4, 5, 2), то есть сечение останавливает процесс после четвертого события процесса p_1 , пятого события процесса p_2 и второго события процесса p_3 . И сечение C_2 имеет границу (5, 6, 3). Сечение представляет собой экземпляр, когда запрос на описание индивидуального состояния принимаются всеми членами группы. Очевидно, не все сечения имеют смысл. Например, сечение C_1 с границей (4, 5, 2) нарушает наше априорное знание о причинности. Оно включает в себя событие e_2^4 , вызванное получением сообщения m_3 процессом p_2 , но не включает в себя событие e_3^3 (отправка сообщения m_3), вызванное процессом p_3 . На этом снимке работы бота процесс p_3 был приостановлен после второго события e_3^2 , прежде чем у него возникла возможность отправить сообщение m_3 , то есть причинно-следственная связь нарушается и система никогда не сможет достичь глобального состояния. Сечение $C_2 = (e_1^5, e_2^6, e_3^3)$ согласованное, так как нет причинно-следственного несоответствия, оно включает в себя событие e_2^6 (отправка сообщения m_4), но не включает в себя причинные следствия от него – событие e_3^4 (получение сообщения m_4 от процесса p_3).

Далее введём понятия согласованных и несогласованных сечений. Замкнутое сечение, подчиненное причинно-следственным приоритетам отношений, называется согласованным сечением. C является согласованным сечением для всех событий тогда и только тогда

$$\forall e, e', (e \in C) \wedge (e' \rightarrow e) \Rightarrow (e' \in C). \quad (9)$$

Используя согласованное сечение можно определить, произошло ли событие e до сечения.

Согласованный запуск событий R – общая упорядоченность событий, входящих в запуск, согласуется с частичным порядком, навязанным причинной-следственной связью. Для всех событий $e \rightarrow e'$ подразумевается что в запуске событий R , событие e появится до события e' .

Рассмотрим информационный поиск студента, состоящие из группы общающихся между собой процессов $G = \{p_1, p_2, \dots, p_n\}$. История $\gamma(e)$ причинно-следственного события e , является наименьшим согласованным сечением группы процессов G , включающим в себя событие e .

$$\gamma(e) = \{e' \in G | e' \rightarrow e\} \cup \{e\}. \quad (10)$$

Причинно-следственная история события e_2^5 это наименьшее согласованное сечение включающее событие e_2^5 :

$$\gamma(e_2^5) = \{e_1^1, e_1^2, e_1^3, e_1^4, e_1^5, e_2^1, e_2^2, e_2^3, e_2^4, e_2^5, e_3^1, e_3^2, e_3^3\}. \quad (11)$$

Это самый маленький согласованный срез, включающий в себя событие e_2^5 . Действительно, если опустить событие e_3^3 , то срез (5, 5, 2) будет противоречивым и

несогласованным – он будет включать коммуникационное событие e_2^4 для приема сообщения m_3 , но не будет включать событие e_3^3 – отправка сообщения m_3 . Если опустить событие e_1^5 , то срез (4, 5, 3) также будет несогласованным, так как будет включать в себя событие e_2^3 , но не будет включать в себя событие e_1^5 .

Следующий алгоритм может быть использован для построения причинно-следственных истории:

Каждый процесс $p_i \in G$ начинается с $\Theta = \emptyset$.

Каждый раз когда процесс p_i получает сообщение от процесса p_j он создает историю:

$$\gamma(e_i) = \gamma(e_j) \cup \gamma(e_k) \quad (12)$$

где, e_i – коммуникационное событие получения сообщения; e_j – предыдущее локальное событие процесса p_i ; e_k – коммуникационное событие отправки сообщения процессом p_j .

К сожалению, такое объединение историй порождает слабоуправляемый прирос больших данных, так как причинно-следственные истории растут очень быстро.

Список литературы

1. Примеры кода бота. [Электронный ресурс] URL: <https://core.telegram.org/bots/samples> (дата обращения: 29.11.2017).
2. Маринеску Д. К. Теория и практика облачных вычислений [Текст]. США: Морган Кауфманн – издано в Элсевера, 2013. 415 с.

"FEMULAS" INTERACTIVE MOTIVATOR OF STUDYING THE SUBJECT

S.Yu. Petrova | Novgorod State University
Cand. Sci. (Engineering), associate professor
svetlana.petrova@novsu.ru
Velikiy Novgorod

Abstract. The article discusses the using Famulus - new information technologies in education. The modern education system faces problems of improving the technologies for substantive work of students and organizing inclusive education for disabled people, especially if the student is poorly motivated in training. It turned out that most students are not able to determine the value of the publication and independently find the necessary knowledge on the subject matter. Thus, there was formulation of the task: we need to develop an interactive motivator that makes it possible to simplify as much as possible a selection of relevant, interesting and qualitative information on the subject under study. Data collection for the motivator is gathered by a bot, whose main function is to detect events of interaction of the expert in the student's study, the subject area with digital educational resources of the NovSU portal and the Internet. The data obtained are analyzed by the Famulus technology solution, which forms a global history of cause and effect relations between the expert and digital educational resources and determines the importance of the page. During the phase of the specification of cause-effect relationships, the user's transition to digital resources, we view as a finite state machine and determine the actions that cause transitions from one digital resource to another. The causal relationship of local events can be obtained from the history of the process. Two events in the global history can be connected. If this is so, none of them is the cause of the other, so we can say that such events are parallel events. The received

histories go through a procedure for processing big data by the Hadoop cluster at NovSU. The result of processing will be a list of links to http pages with relevant, interesting and high-quality publications on the subject under study.

Keywords: information technology in education, big data, expert systems, personalization of education

References

1. Bot Code Examples URL: <https://core.telegram.org/bots/samples> (Accessed: 29.11.2017).
2. Marinescu, D. C. (2013) Cloud Computing Theory and Practice. USA: Morgan Kaufmann is an imprint of Elsevier. 415 p.

УДК
004.912,
51-74

СТАТИСТИЧЕСКИЕ МОДЕЛИ ЯЗЫКА

<p>Мурат Казбекович Датиев аспирант г. Владикавказ</p>	<p>Московский институт радиотехники, электроники и автоматики</p>
<p>Александр Юрьевич Кулай старший научный сотрудник г. Владикавказ</p>	<p>Московский институт радиотехники, электроники и автоматики</p>
<p>Казбек Муратович Датиев к.т.н., профессор datiev_skgmi@mail.ru г. Владикавказ</p>	<p>Северо-Кавказский горно- металлургический институт</p>

Аннотация. В работе рассматриваются современные статистические модели языка. Дано определение применяемым критериям эффективности моделей. Описываются следующие статистические модели языка: n-граммные модели, модели на основе деревьев решений, лингвистически мотивированные модели.

Ключевые слова: статистические языковые модели, n-граммные модели, перплексия.

Введение. При помощи статистического моделирования языка (Statistical Language Modeling (SLM)) формализуются закономерности естественного языка для улучшения производительности различных естественно-языковых приложений. Статистическое языковое моделирование строит оценку вероятностного распределения различных лингвистических единиц, таких как слова, предложения, тексты [Розенфельд, 1996]. Статистические модели языка применяются в большом количестве приложений: распознавание речи, машинный перевод, классификация документов, информационный поиск, распознавание рукописного текста, проверка орфографии и др.

SLM использует технологии статистической оценки, применяя для обучения языковые данные – текст. Из-за категориальной природы языка и огромного словарного запаса статистические технологии должны оценивать большое количество параметров и, следовательно, сильно зависят от доступности значительного количества обучающих