

ПРАКТИЧЕСКИ ПРИМЕР НА ТЕХНОЛОГИЧНИ СРЕДСТВА ЗА РЕАЛИЗИРАНЕ НА УЕБ 2.0 ОБУЧЕНИЕ ПО МАТЕМАТИКА

Филип Петров, Даниела Дурева-Тупарова

Резюме. В статията са представени софтуерни приложения за интегриране на технологии за изобразяване на математически формули и въвеждане на математически текст в уеб форми. Поставен е акцент върху избор на технологични средства, които не изискват инсталиране на специализиран софтуер на сърверните системи. Това помага на разработчиците на съществуващи сайтове да преминат по-лесно към уеб 2.0 обучение без да има нужда от преконфигуриране на вече работещи системи, което е важен фактор за сайтове със споделен хостинг на сървери в университетски или училищни мрежи и обичайна практика администрирането да бъде изключително консервативно. Специално внимание е отделено на възможностите за прехвърляне на информация между уеб сайта и други приложения, както и възможностите за индексирание на въведения математически текст от търсещи машини.

Keywords: web 2.0 обучение, latex, mathml, обучение по математика

ПРЕДИМСТВАТА НА УЕБ 2.0 ОБУЧЕНИЕТО

В педагогическата литература задълбочено са изследвани традиционните форми на обучение - фронтална, групова и индивидуална (Тодорина, 2011). С развитието на информационните технологии се появяват нови възможности за тяхната реализация. Уеб 2.0 обучението си поставя за цел да се направи електронен модел на традиционните форми на обучение, с което да се дадат допълнителни средства за подпомагане на самостоятелната подготовка на ученици и студенти и да се постигне ефективно дистанционно обучение. В (Петров, 2010) е обобщена следната дефиниция за Уеб 2.0 обучение: „Уеб 2.0 базирано обучение наричаме електронен ресурс в Интернет, който е интерактивен носител на учебна информация, поднесена във вид на текст, графики, видео и друга мултимедия, в който учениците обменят опит помежду си и допълват учебните материали, водейки дискусии под различна форма с учителя и/или помежду си”.

Основно и важно изискване за едно електронно обучение да бъде класифицирано като „уеб 2.0“ е комуникацията между участниците в него (независимо от нейната форма) да бъде интегрирана като компонент от самото учебно съдържание. Погрешно е да се смята, че в уеб базираното обучение няма субект-субектна комуникация. Напротив, във вече масово навлязлото в практиката уеб базирано обучение също има наличие на комуникация между участниците в обучението. Разликата между „уеб“ и „уеб 2.0“ обучението не е наличието или отсъствието на комуникационни канали,

а то е „вплитането“ на тези комуникационни канали като неделима част от учебния процес. По този начин се намаляват основните негативни ефекти от липсата на физически контакт между участниците в обучението, а се запазват всички предимства от традиционното уеб обучение, описани в (March, 1996), (Aggarwal, 2003), (Тупаров & Дурева, 2008) и др. Възможността за допълване на учебните материали от обучаемите е допълнително и много мощно средство за подобряване на качеството на материалите, разбира се след задължителна редакция и обобщение от страна на преподавателя.

ОСНОВНИ ПРОБЛЕМИ НА УЕБ 2.0 ОБУЧЕНИЕТО ПО МАТЕМАТИКА

Учебното съдържание по математика се отличава с наличието на много формули, знаци и символи, които не е възможно да бъдат написани като обикновен текст чрез стандартна компютърна клавиатура и без наличието на специализиран софтуер. Поради тази причина авторите на учебно съдържание по математика отдавна са свикнали с употребата на технологични средства като MathType (Campbell, 1991) и редактори за Latex (Gratzer, 1995). В уеб и уеб 2.0 обучението по математика обаче се появяват два принципно нови проблема:

- Студентите също трябва да въвеждат математически текст – както при въвеждането на отговори на задачи при интерактивните обучаващи системи, така и в каналите за субект-субектна комуникация, като например чат, форуми, социални мрежи и др., подробно представени в (Петров, 2010);

- Математическите формули трябва да бъдат изобразявани в уеб страници по начин, по който лесно да могат да бъдат копирани и пренасяни в други софтуерни приложения (например текстови редактори като Word) и обратно, както и да бъдат индексирани от търсещи машини на информация.

Съществуват множество решения за писане на математически текст със средствата на специализирани редактори, които са инсталирани на потребителския компютър. Един подход за решаване на първия проблем би бил формулите да се пишат с такъв редактор, след което генерираните файлове да бъдат предавани през комуникационните канали. Това за съжаление би било неефективно от гледна точка на провеждане на дискусия, в която математическите формули се използват често. Ето защо, за целите на Уеб 2.0 обучението, технологиите за въвеждане на математически текст трябва да се развиват към максимална достъпност и интуитивност чрез интегриране директно в уеб форми. Колкото до втория проблем – неговото решаване отдавна търпи развитие чрез появата на редица технологични средства за визуализация на Latex код и дори нови стандарти за писане на код в уеб страници като MathML, но едва в последните години се забеляза значим прогрес в областта.

В (Петров, 2011) е проведено е изследване със сравняване на 13 популярни технологични средства за въвеждане и изобразяване на математически текст в уеб страници. За сравнително краткия период от тогава досега, технологиите претърпяха ново качествено развитие. При избора на технологични средства за писане

и визуализиране на математически текст в уеб форми могат да се имат предвид следните изисквания:

- Да бъдат безплатни и със свободен лиценз за използване;
- Потребителите на системата трябва да могат самостоятелно или с минимална помощ да усвоят тези технологии;
- Да поддържат (при това напълно) широко разпространен стандарт за писане на математически формули на компютър, за да може да се осигури лесно преизползване на вече написани текстове;
- Да има възможност формули, които са визуализирани в уеб сайта, да бъдат копирани и вмъквани в други редактори (т.е. да има възможност за връзка с други технологични средства), като това да не изисква специални права за достъп на потребителите;
- Да не се изисква инсталиране на допълнителен софтуер както на потребителските компютри, така и на сървера – по този начин ще се гарантира лесно интегриране на тези технологии в съществуващи образователни сайтове и системи, без да се правят съществени промени по конфигурацията.

Базирайки се на посочените изисквания, са подбрани за по-подробно представяне две средства, които в комбинация помежду си ги удовлетворяват – MathJax и CodeCogs Equation Editor.

MATHJAX 2.1

MathJax¹ е технологично средство, написано на JavaScript, което позволява въвеждане и изобразяване на математически текст в уеб страници. В публикацията (Петров, 2011) е отбелязано, че то е един от лидерите в тази област, но страда от няколко недостатъка, основният от които е значително по-бавното зареждане спрямо неговите конкуренти. На 26 февруари 2012 г. излиза нова, силно подобрена версия 2.0, която практически разрешава този проблем. Към момента на написването на статията са налични още подобрения във версия 2.1. Софтуерът се поддържа от всички популярни уеб браузъри, включително и редица мобилни устройства.

Интегрирането на MathJax 2.1 към съществуваща уеб страница е изключително лесно и се състои в две стъпки. Първата стъпка е да се добави javascript файла на MathJax в областта <head> на уеб страницата. Това се получава чрез добавяне на следния код:

```
<script type='text/javascript' src =  
,http://cdn.mathjax.org/mathjax/latest/MathJax.js?config=TeX-AMS-MML_HTMLorMML '  
></script>
```

Чрез този код се добавя файлът MathJax.js от CDN сървера на MathJax (позволено е и да се свалят файловете на MathJax в директория на сървера, на който е инсталиран сайтът и да се зареждат локално). Допълнителните настройки указват

кои технологии ще могат да бъдат използвани на сайта (в примера са заредени всички възможни). Втората стъпка е да се добавят специфични за уеб страницата настройки чрез подобен на следния код (отново се добавя вътре в таг <head> на страницата):

```
<script type='text/x-mathjax-config'>
MathJax.Hub.Config({tex2jax: {inlineMath: [[,[texi]','[/texi]']],
                                   displayMath: [[,[tex]','[/tex]']]   }
                    })
</script>
```

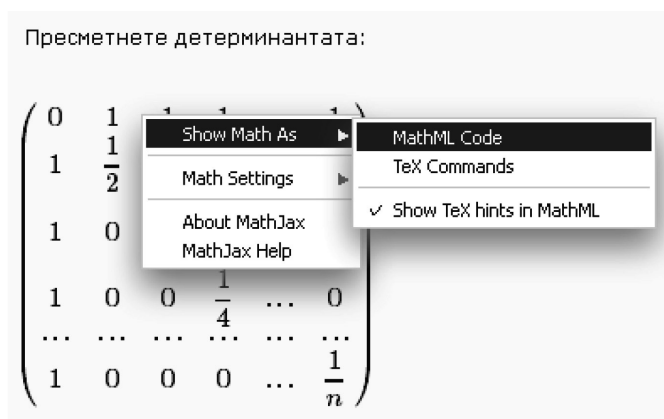
След като това бъде направено, всяка математическа формула, написана на LaTeX, която присъства в текста на страницата и е оградена с таговете [texi] и [/texi] или [tex] и [/tex], ще бъде визуализирана. При това в сорс кода на страницата формулата ще остане като чист LaTeX код, което отговаря на условието да може да се индексира от търсещи машини. Относно визуализацията, шрифтът и големината му се подбират автоматично спрямо текста в параграфа, в който се намира самата формула.

Разликата между “inlineMath” и “displayMath” е, че при “inline” големината на формулата ще бъде скалирана така, че да може да се събере в рамките на един ред в ограждащия я текст – така тя може да се вмъква вътре в нормален текст и да не попречи на форматирането на параграфа. При “display” формулата винаги ще бъде в нормалната големина на шрифта и на нов ред. Що се отнася до копирането на визуализирани формули и пренасянето им в друг софтуер, това може да бъде осъществено чрез щракване с десния бутон на мишката върху формулата, както е показано на фиг. 1.

Копираният като MathML код може да бъде вмъкнат директно в Microsoft Word 2007 или по-нова версия (текстовият редактор ще го конвертира автоматично към формат за MathType) или OpenOffice, а копираният TeX код е напълно съвместим с редактори на LaTeX.

CODECOGS EQUATION EDITOR (v3)

Основен проблем при писането на формули на LaTeX е, че се изискват знания за специфичния синтаксис, на който се въвеждат командите. Това отнема допълнително време за обучение на учениците/студентите, което естествено би било хубаво, ако може да бъде спестено или поне максимално намалено. На помощ идват визуалните редактори за LaTeX код. Съществуват много такива самостоятелни приложения, но тези, които могат да се интегрират в уеб страница, все още са прекалено малко. Такива например са Connexions³, DragMath⁴, FireMath⁵ и Wiris⁷ за MathML и ZohoWriter⁶ и Wiris⁷ за LaTeX. Повечето обаче не отговарят на поставените по-горе изисквания. Те имат различни недостатъци – нямат достатъчно добра поддръжка в различни Интернет браузъри или изискват инсталация на допълнителен софтуер



Фиг. 1. Извличане на код от визуализирана формула

на сървера/клиентската машина (Петров, 2011). От всички изброени Wiris (добил популярност около лесната си интеграция с Moodle) се доближава максимално до поставените изисквания, но не е безплатен. Редакторите Connexions и ZohoWriter също не са безплатни, DragMath не е достатъчно пълноценен откъм възможности за писане на по-сложни формули и изисква инсталиране на Java, а FireMath не работи с различни от Firefox браузъри. Затова фокусът на статията е поставен върху един „смесен“ редактор - CodeCogs Equation Editor², в който се редактира код на LaTeX в текстови вид чрез възможността за вмъкване на шаблони (заготовки) чрез визуални контроли.

Представеният редактор се зарежда автоматично като скрипт от уеб сайта на разработчиците. Позволява се както интегриране към съществуващи редактори (като например популярния TinyMCE използван масово в интернет форумите), така и зареждане в нов прозорец (рорър). Вторият начин за интеграция е съществено по-лесен, защото не изисква сериозни познания по HTML. Първата стъпка е да се вмъкне зареждащият скрипт в областта <head> на страницата чрез следния код:

```
<script type="text/javascript" src="http://latex.codecogs.com/editor3.js"></script>
```

Втората стъпка е да се вмъкне картинка или текст с връзка вътре в тялото на страницата, при последване на която ще се зареди въпросният прозорец. Това може да се направи например по следния начин:

```
<a href="javascript:OpenLatexEditor(
    ,BOXID','phpBB','bg-bg',false,','full'
);">Вмъкни формула</a>
```

Както се вижда, извиква се функция OpenLatexEditor, на която се подават 6 параметъра. В първия параметър трябва да се подаде BOXID, което е името на идентификатора на текстовото поле, в което се въвежда текстът (взима се от <textbox id=... на текстовото поле, в което ще се пише на страницата). Този параметър е

изключително важен, защото указва на редактора къде да вмъкне формулата и съответно къде да върне фокуса при натискане на бутона за вмъкване на кода.

Вторият параметър е името на тага, който ще огради текста. В случая в примера е представен параметър с име “phpBB”, понеже той ще генерира таг [tex] и [/tex], което беше избрано и в примера за инсталация на MathJax. Други възможности за този параметър са представени в сайта на производителя и са следните:

<i>type</i>	Example Output
latex	$1+\sin(x)$ or $\backslash 1+\sin(x)\backslash$
doxygen	$\f$1+\sin(x)\f$ or \f[1+\sin(x)\f]$
phpBB	[tex]1+\sin(x)[/tex]
tw	[img[http://www.codecogs.com/gif.latex?1+\sin(x)]]
html	
pre	<pre xml:lang=“latex”>1+\sin(x)</pre>

Третият параметър контролира езика. Българската версия е преведена с помощта на първия автор на настоящата статия. Четвъртият контролира самия редактор и това дали кодът в него ще бъде генериран на един ред или може да бъде на няколко (препоръчително е кодът да е на един ред, защото в противен случай може да се появят проблеми с визуализацията от страна на MathJax). Петият параметър (който в примера е празен) е възможност за добавяне на код на LaTeX, който да бъде зареден по подразбиране в редактора (това е удачно тогава, когато в процеса на обучение се решават типови задачи). Последният параметър контролира дизайна на редактора. Стандартните стойности са “mini” и “full”, като разликата е в броя на бутоните в менюто. Реално редакторът позволява да бъдат добавяни и премахвани всякакви бутони по усмотрение⁸.

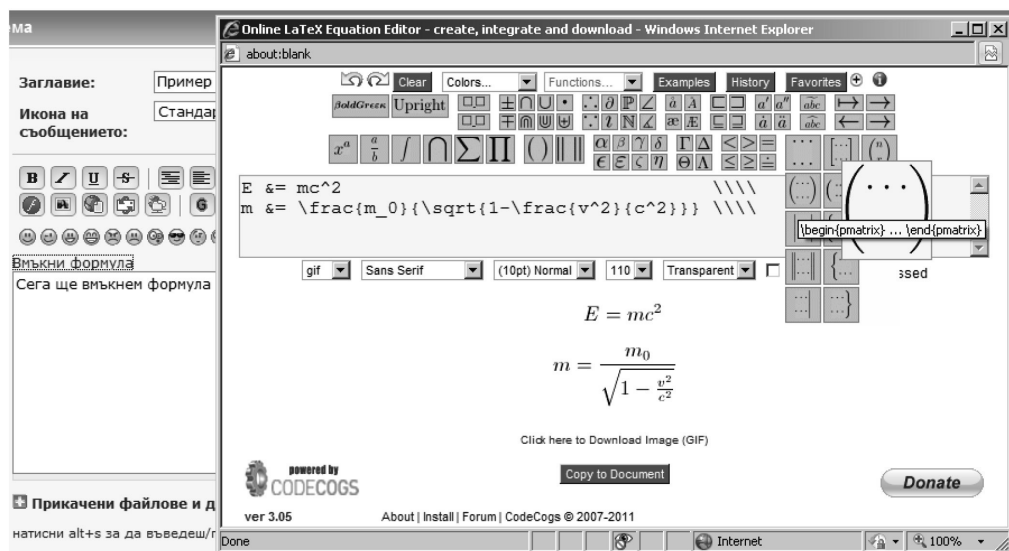
При показаната конфигурация редакторът ще се отваря в нов прозорец, след което ще може да се използват визуалните контроли, показани с пример на фиг.2. При натискането на бутон от менюто ще бъде генериран автоматично шаблон на LaTeX. При попълването на шаблона ще се визуализира картинка под текстовото поле, която в реално време показва как изглежда формулата. При натискане на бутона “Copy to document” кодът от редактора ще бъде пренесен в текстовото поле на страницата на сайта, като ще бъде ограден автоматично от съответния таг (в примера [tex] и [/tex]).

Както се вижда от примера, редакторът не е изцяло визуален (каквито например са MathType в Word или Connexions, DragMath и Wiris за уеб страници). Това може да бъде отчетено както като голям недостатък, ако редакторът трябва да бъде използван при обучение с ученици и студенти, които нямат сериозна подготовка за работа с компютър и нямат опит с писане на код. От друга страна, при обуче-

нието на студенти по математика, информатика и технически науки, както и при обучение на ученици от математически гимназии, това може да бъде отчетено и като предимство, понеже по интуитивен начин (с шаблони) запознава потребителя с писането на код на LaTeX. Това са допълнителни знания, които не са излишни в обучението по математика.

ЗАКЛЮЧЕНИЕ

Представените две технологични средства могат успешно да се интегрират във всякакви уеб сайтове, без да се налагат сериозни промени по техния програмен код. MathJax към момента е безспорен лидер в областта на визуализирането на математически формули в уеб страници. CodeCogs Equation Editor може да бъде използван при обучение на студенти по математика, информатика и технически науки или при ученици от гимназиална степен с изявени интереси в областта на математиката и информатиката. За ученици и студенти с хуманитарна насоченост е по-подходящо технологичното средство за въвеждане на математически текст да бъде изцяло визуално и да не изисква познания по писане на програмен код, следователно ще бъде по-подходящо да се подбере друго технологично средство за въвеждане на математически текст в уеб форми.



Фиг. 2. Пример за написване на формули в CodeCogs Equation Editor

Използването на web 2.0 средства и в частност интегрирането на приложения за писане на математически символи разширява възможностите не само за създаване и предоставяне на учебно съдържание, но и за проверка и оценка във web 2.0 среда на математически компетенции, които са част от ключовите компетентности.

Благодарности:

Изследването е подпомогнато от Фонд „Научни изследвания“, проект № ДФ-НИ-И01/10

БЕЛЕЖКИ

1. MathJax, <http://www.mathjax.org> (последно достъпена на 21.09.2012г.).
2. CodeCogs Equation Editor, <http://www.codecogs.com/latex/about.php> (последно достъпена на 21.09.2012г.).
3. Connexions MathML Editor, <http://cnx.org/matheditor> (последно достъпена на 21.09.2012г.).
4. DragMath, <http://www.dragmath.bham.ac.uk> (последно достъпена на 21.09.2012г.).
5. FireMath, <http://www.firemath.info> (последно достъпена на 21.09.2012г.).
6. ZohoWriter, <http://writer.zoho.com> (последно достъпена на 21.09.2012г.).
7. Wiris, <http://www.wiris.com> (последно достъпена на 21.09.2012г.).
8. CodeCogs Design, <http://www.codecogs.com/latex/design.php> (последно достъпена на 21.09.2012г.).

ЛИТЕРАТУРА

- Петров, Ф. (2010). Уеб 2.0 обучение по математика, *сборник “Синергетика и рефлексия в обучението по математика”*, Пловдив, университетско издателство “Паисий Хилендарски”, стр. 445-452,
- Петров, Ф. (2011). Технологии за въвеждане и изобразяване на математически формули в Уеб 2.0 приложения, *сборник “Образованието в информационното общество”*, Пловдив, Асоциация “Развитие на информационното общество” и “Институт по математика и информатика” към БАН, стр. 200-209,
- Тодорина, Д. (2011). Форми на обучение, издателство “Санин – Н и Н“ ООД Благоевград
- Тупаров, Г.; Дурева, Д. (2008). Електронно обучение, *Университетско издателство „Неофит Рилски“ Благоевград*,
- Aggarwal, A. (2003). Web based education: learning from experience, *Informational Science Publishing*,
- Campbell, K. (1991). Math type: A reasonable option for mathematical equations and formulas, *American Journal of Distance Education, Volume 5, Issue 1, pages 72 – 74*
- Gratzer G. (1995). Math into Latex: An Introduction to Latex and AMS-Latex, *Birkhauser Boston*,
- March, T. (1996). Working for the web education, *Classroom Connect Conference, Anaheim*.

Филип Петров

✉ асистент

Катедра „Компютърни системи“

Технически Университет, София

<http://www.cphpvb.net>

E-mail: philip@abv.bg

Даниела Дурева-Тупарова

✉ доцент, доктор

Катедра „Информатика“

Югозападен университет „Неофит Рилски“

Благоевград

E-mail: ddureva@swu.bg

PRACTICAL EXAMPLES OF TECHNOLOGICAL TOOLS FOR IMPLEMENTATION OF WEB 2.0 EDUCATION IN MATHEMATICS

Abstract. Software technologies for integration of visualization and editing of mathematical formulas in web forms are presented in the article. The main focus is set on choice of technological resources which do not require installation of specific software on the hosting servers. This helps the conversion of existing web sites into web 2.0 educational resources without the need of reconfiguring of the existing systems. Sometimes this is an important factor for shared hosting, school and university networks which are traditionally very conservative to changes. A special attention is given to the copy/paste capabilities for transferring of mathematical formulas back and forth between the web site and other software products.

Filip Petrov

✉ Assistant

Computer Systems Department

Technical university Sofia

<http://www.cphpvb.net>

E-mail: philip@abv.bg

Daniela Dureva-Tuparova

✉ Docent, PhD

Informatics Department

South-West Universiti “Neofit Rilski”

Blagoevgrad

E-mail: ddureva@swu.bg