[Downloaded from www.aece.ro on Sunday, July 06, 2025 at 00:36:48 (UTC) by 172.69.58.62. Redistribution subject to AECE license or copyright.]

Building a Module for Inserting Microformats into Moodle

Iasmina ERMALAI, Bogdan DRAGULESCU, Andrei TERNAUCIUC, Radu VASIU
Politehnica University of Timisoara, 300006, Romania
Multimedia Centre, Faculty of Electronics and Telecommunications, Timisoara, 300223,Romania
iasmina.ermalai@cm.upt.ro

Abstract—**Information found nowadays on the World Wide Web is generally regarded as limitless. Hence emerged the need to find methods for organizing data in order to get better information retrieval. Organizing web content does not implicitly mean structuring data by attaching meaning to the information published on the web – at least not yet. For instance, content management systems (CMS) are tools that provide means of organizing web data. Nevertheless only a handful of CMSs have a semantic layer. The following paper offers an in-depth description of the logic and the implementation behind the method used to integrate microformats – also known as the "lower-case semantic web"– into the widely known learning content management system Moodle, personalized to suit the needs of students attending the Politehnica University of Timisoara. It is an easy to use, easy to integrate solution that does not require any changes in the core of Moodle and, moreover, it does not overrun the server, as the decision block runs on the users' computer.**

Index Terms—**content management, electronic learning, Information representation, semantic Web, Web services.**

I. Introduction

The use of intelligent applications aims to offer methods of collecting information from various web resources, processing it, and exchanging results between them, all these in order to facilitate and enhance the user experience. Yet, all these processes rely heavily on the particular methods through which information is published on the web. Information should receive a well defined meaning, a structure that would allow a better communication between machines and software, therefore enhancing interactions between humans and computers. Since the information and applications available nowadays on the World Wide Web are countless, a complete WWW reconstruction appears at this time, at best, a herculean task. The easier, more feasible alternative would be to add a semantic layer to the information, in order to extend functionalities of existing applications. This extension of the current web is known as the Semantic Web, as envisioned by Sir Timothy Berners-Lee [1]. Semantic web publishing tools, context modelling tools, and semantic search engines are all technologies and tools which are already in use.

Although built with the same goals as the Semantic Web – to structure the web content, to "support decentralized knowledge management"– microformats vary in several aspects: they "do not address *implicit* knowledge representation, ontological analysis, or logical inference", though they can "encode *explicit* information to aid machine readability" [2]; they "are for human consumption first, machine readability second", as they imply "incorporating markup into existing web documents over the creation of new formats (e.g. OWL, RDF)" [3]. Microformats are simple, yet structured methods, based on existing standards (HTML and CSS), used for adding more meaning to web pages, in order to better indicate people, companies, events, reviews, tags, and so on [4] [5].

Content management systems or CMSs are different methods of organizing content on the web, most of them lacking a semantic layer. They are widely used in various fields, from eLearning to eCommerce, from eHealth to eGovernment.

One of the most used CMSs in education (currently around 68,000 registered sites, from 220 countries [6]) is Moodle, an open source online learning system, which allows universities to manage content, students and activities. The initiatives to include Semantic Web technologies into Moodle have been rather scarce, difficult and mostly in a prototype phase, thus not available to the community of users.

In 2009, the "Politehnica" University of Timisoara made the first steps towards adding semantics into the online portal used at that time for managing the pedagogical activity [7]. Seeing that, in 2010, the University, first partially and then completely replaced the mentioned portal with Moodle (two separate instances: Virtual Campus - http://cv.upt.ro/ and ViCaDiS – Virtual Campus for Digital Students - http://www.vicadis.net/campus/), the focus switched towards adding microformats into the well known LCMS, initiative started in 2011 [8]. This paper presents an insight into the logic and implementation behind the latest version of microformats' integration into Moodle. The novelty resides in the easy to integrate module that does not affect the core of Moodle and does not overload the server.

The present paper is organized as it follows: first, issues related to intelligent Web, also known as the Semantic Web, and to learning content management systems are addressed. Second, the paper formulates the problem we are trying to solve, followed by the proposed solution, described in the third section. The fourth section presents the discussions related to the suggested approach, while the last section offers a few future perspectives of the current work.

This paper was supported by the project "Development and support of multidisciplinary postdoctoral programmes in major technical areas of national strategy of Research - Development - Innovation" 4D-POSTDOC, contract no. POSDRU/89/1.5/S/52603, project co-funded by the European Social Fund through Sectorial Operational Programme Human Resources Development 2007-2013.

Digital Object Identifier 10.4316/AECE.2013.03004

[Downloaded from www.aece.ro on Sunday, July 06, 2025 at 00:36:48 (UTC) by 172.69.58.62. Redistribution subject to AECE license or copyright.]

II. FORMULATING THE PROBLEM

Moodle is one of the most used learning content management systems. Even if open source, it offers a fair amount of built-in tools that provide online interaction/collaboration between students and tutors, course delivery, and activity management. There is also a considerable amount of modules and blocks developed by the Moodle community.

The fact that the support offered for open source solutions was rather poor and extra features, customized for particular needs, had to be developed in-house, were issues often arisen when using Moodle. Furthermore, attempts to attach semantic reasoning to Moodle were rather scarce and were mostly focused on metadata, adding an additional semantic layer, and data mining, which usually generate results in the RDF (Resource Description Framework) format [9] [10]. Moodle 2.0 was also used for creating the Social Semantic Web for Lifelong Learners, a “a dynamically personalized learning environment for the lifelong learner” [11]. A different Semantic approach to Moodle consisted of creating a plug-in that allowed “semantic recovery of tweets and resources from external repositories” [12]. Other initiatives used intelligent agents to process the information previously labeled using various ontologies, found in the Moodle semantic layer. [13]

mEducator 3.0[2] is a contemporary, more complex project that aims at using Semantic Web to add a structure to existing medical standards and materials. It is “is a sharing tool for medical educational content, based on the principles of open linked data” [14] and it includes solutions based on Elgg, Drupal, Semantic Media Wiki and Moodle. [15] The prototype module developed for Moodle intended to offer users the possibility to “search, retrieve, reuse and repurpose medical resources with the minimum of extra effort”. The user could also “retrieve and incorporate the intended digital resource into his course, easily and without leaving the LCMS” [16].

The purpose of this paper is to describe a simple, easy to use solution, for integrating microformats into Moodle. Its aim is not to enhance learning, but merely to offer the user an alternative solution to deal with administrative tasks related to contacts and schedule. The only extra requirements, necessary in order to be able to use microformats for their administrative feature, are the installation of a browser plug-in and some skills in configuring and using it.

Furthermore, there is an important additional advantage, that should not be overlooked, namely the ever increasing use of open standard like RDFa, Microformats and Microdata [17] by search engines like Google, in order to display Rich Snippets [18]. Rich Snippets are “a new presentation of snippets”, that rely on markup formats and Google algorithms, to “give users convenient summary information about their search results at a glance” [19]. This significant feature made available through microformats and search engines’ algorithms, had a considerable weight in our decision to publish parts of the web information from Moodle learning system using hCard (for tutor and student contact information) and hCalendar (for events).

III. PROPOSING A SOLUTION

The insertion of microformats inside the pre-existing HTML structure (provided by the platform) is done with minimal intervention upon Moodle’s core code. Thanks to its signature modular make-up, we were able to create a custom block inside its file structure, containing all the necessary scripts, which would not be overwritten in the case of a system update. Additionally, since the theme used for our Moodle installation is also custom-made, the script called on each page load is also safe from tampering by the update process. The relation between the different types of scripts is described in Fig. 1.

As shown in the block diagram, it all begins with a decision block. This is a small script written with the aid of the popular JavaScript library jQuery and inserted into the header file of the template. This means that, for each and every page Moodle produces, these few lines of code will also be loaded. This is the only modification of code outside of the Moodle block we specifically created, but is essential in order to ensure that microformats are created for all users, regardless of their privileges, capabilities or customizations.

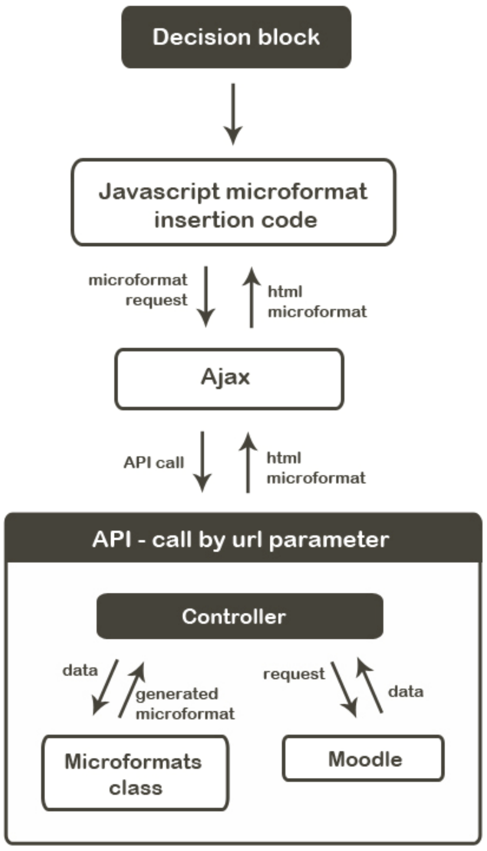


Figure. 1: Block diagram of the microformats’ creation algorithm

What this block does is to check the URL of the current page and extract two pieces of information. The first is the name of the actual page that was loaded. This way the algorithm checks which Moodle section the user is accessing and decides whether one or the other microformat needs to be inserted into the page. For instance, if the user is checking the calendar, the block extracts the “calendar/view.php” part of the URL and decides to call the hCard injection algorithm. If this first piece of information does not correspond to any of the predefined cases in which

[Downloaded from www.aece.ro on Sunday, July 06, 2025 at 00:36:48 (UTC) by 172.69.58.62. Redistribution subject to AECE license or copyright.]

the microformats are needed, the script does nothing.

The second piece of information extracted from the url are the parameters that customize the view. In the above example, these parameters decide if the user is accessing a whole month or only the events from one day, and also, which month and which day. This is essential for the algorithm, since we need to provide the users with only the events from the desired period, and not all of the events in the database.

As stated above, this code snippet is called in each page of the platform. This may seem like an "overkill" and a strain on resources, but since this is JavaScript, all the necessary computing power comes from the user's computer, and not from the server. Secondly, if the decision taken by the code is not to call on any microformat, the execution of the script ends here, with a minimal execution time and processing. The advantages of this approach (the fact that we didn't modify the main code of Moodle, and in doing this, we insured a greater portability and flexibility in further developing the scripts) far outweigh the few extra milliseconds needed to load each page.

Once the decision to insert a microformat has been taken, the script will make an Ajax call to an API located inside the custom block in Moodle. This call includes, aside from the correct target (each of the two microformats implemented here have their own controllers), the parameters from the URL. By using jQuery, this Ajax call has been reduced to a couple of lines of code, greatly improving the compatibility and also the efficiency of the process.

The API will return the code that needs to be injected into the page. This code represents the microformat backbone, a simple HTML code with specific classes and structure that will then be read by the microformat interpreter which the users need to have installed on their computers. For testing purposes, we have used the Mozilla Firefox add-on Operator.

The code is inserted using jQuery's extensive DOM manipulation capabilities. In order not to interfere with the existing code, we have created a dummy container, in essence a DIV tag that we positioned outside of the visible area of the document, and which is removed a short time after. This delay and the fact that we didn't just use an invisible container (with "*display:none;*" in the style section) instead of positioning it outside the page, were necessary for the microformats interpreter, which needs a little time in order to "scan" the page for the required code.

This functionality (the appropriate Ajax calls and the insertion of the code returned by these calls) adds a few lines to the script inside the header of the template. This brings the total to a mere 20 lines of code to be added to the template header.

The rest of the code is found in the custom block we created inside the "blocks" section of Moodle's files. There are two controllers, two classes and two template files, one for each microformat.

Inside the API, the controller gets the parameters provided by the Ajax script and splits them into chunks it can understand. For instance:

?view=month&course=1&cal_d=1&cal_m=05&cal_y=2011

becomes:

- *view=month* - this means that the controller needs to get the events from a whole month;
- *course=1* - this is the ID of the Moodle course (in this case, the whole site, but it can differ according to where the user calls the calendar from);
- *cal_d=1* - this is the day of the month to be viewed; in "month view", this will always be 1, but in "day view", it differs accordingly;
- *cal_m=05* - the month to be shown is the month of May;
- *cal_y=2011* - this is the year of the requested events.

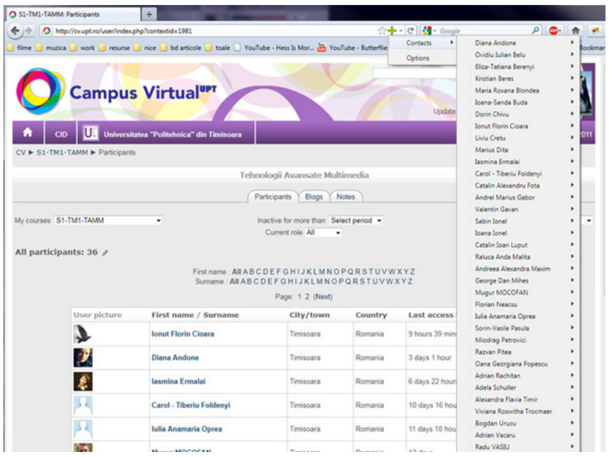
The controller interprets all this information and establishes the period of time for which it must search inside the database for events. For this, it uses Moodle's own functions to get the records from the database, also benefiting from the internal roles-capabilities system which insures that users can access only the events they are allowed to view.

Once the data is retrieved, the controller calls upon the appropriate class and provides it with the information from Moodle. In return, it will get the generated microformat, in its final form.

This microformat will then be returned to the Ajax call, where the initial script, the one from the header, will inject it into the DOM.

And that is pretty much it.

The pictures below show the front end of the functionality added to Moodle and described above:


S1-TM1-TAMM: Participants
http://cv.upt.ro/user/index.php?contextid=1981
filme
muzica
work
resurse
nice
bd articole
toale
YouTube - Hess Is Mor...
YouTube - Butterflie
Contacts
Options
Campus Virtual UPT
CID
Universitatea "Politehnica" din Timisoara
CV ▶ S1-TM1-TAMM ▶ Participants

Tehnologii Avansate Multimedia

Participants | Blogs | Notes

My courses: S1-TM1-TAMM

Inactive for more than: Select period

Current role: All

All participants: 36

First name : All A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

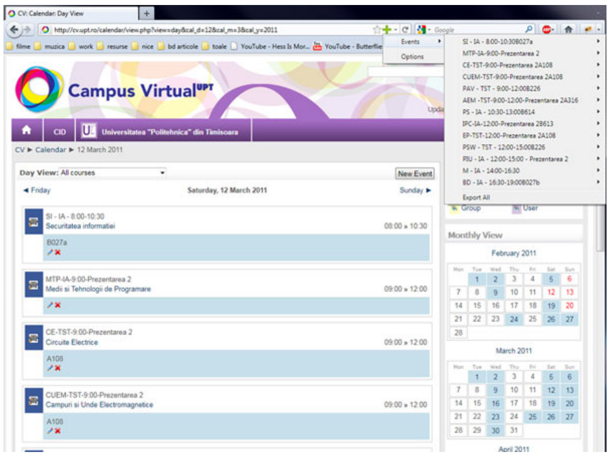
Surname : All A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Page: 1 2 (Next)

User picture	First name / Surname	City/town	Country	Last access
	Ionut Florin Cioara	Timisoara	Romania	9 hours 39 min
	Diana Andone	Timisoara	Romania	3 days 1 hour
	Iasmina Ermalai	Timisoara	Romania	6 days 22 hour
	Carol - Tiberiu Foldenyi	Timisoara	Romania	10 days 16 hou
	Iulia Anamaria Oprea	Timisoara	Romania	11 days 18 hou

Diana Andone
Ovidiu Iulian Belu
Eliza-Tatiana Berenyi
Kristian Beres
Maria Roxana Blondea
Ioana-Sanda Buda
Dorin Chivu
Ionut Florin Cioara
Liviu Cretu
Marius Dita
Iasmina Ermalai
Carol - Tiberiu Foldenyi
Catalin Alexandru Fota
Andrei Marius Gabor
Valentin Gavan
Sabin Ionel
Ioana Ionel
Catalin Ioan Luput
Raluca Anda Malita
Andreea Alexandra Maxim
George Dan Mihes
Mugur MOCOFAN
Florian Neacsu
Iulia Anamaria Oprea
Sorin-Vasile Pasula
Miodrag Petrovici
Razvan Pitea
Oana Georgiana Popescu
Adrian Rachitan
Adela Schuller
Alexandra Flavia Timir
Viviana Roswitha Trocmaer
Bogdan Urucu
Adrian Vacaru
Radu VASIU

Figure. 2: Operator plug-in detecting contact information published with hCard



Day View: All courses

Saturday, 12 March 2011

SI - IA - 8:00-10:30
Securitatea informatiei 08:00 » 10:30
B027a

MTP-IA-9:00-Prezentarea 2
Medii si Tehnologii de Programare 09:00 » 12:00

CE-TST-9:00-Prezentarea 2
Circuite Electrice 09:00 » 12:00
A108

CUEM-TST-9:00-Prezentarea 2
Campuri si Unde Electromagnetice 09:00 » 12:00
A108

SI - IA - 8:00-10:30B027a
MTP-IA-9:00-Prezentarea 2
CE-TST-9:00-Prezentarea 2A108
CUEM-TST-9:00-Prezentarea 2A108
PAV - TST - 9:00-12:00B226
AEM -TST-9:00-12:00-Prezentarea 2A316
PS - IA - 10:30-13:00B614
IPC-IA-12:00-Prezentarea 2B613
EP-TST-12:00-Prezentarea 2A108
PSW - TST - 12:00-15:00B226
PIU - IA - 12:00-15:00 - Prezentarea 2
M - IA - 14:00-16:30
BD - IA - 16:30-19:00B027b
Export All

February 2011

Mon	Tue	Wed	Thu	Fri	Sat	Sun
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28						

March 2011

Mon	Tue	Wed	Thu	Fri	Sat	Sun
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

April 2011

Figure. 3: Operator plug-in detecting events published with hCalendar classes

[Downloaded from www.aece.ro on Sunday, July 06, 2025 at 00:36:48 (UTC) by 172.69.58.62. Redistribution subject to AECE license or copyright.]

IV. DISCUSSIONS

We are not debating here the pros and cons of microformats. Despite their relative maturity as a concept, they have failed to catch on as a mainstream “must have” technology. For instance, all the major browsers include in their latest versions RSS readers. For microformats, you need to install add-ons. But for those who do use them, they provide an easy to use and easy to integrate solution for standard, everyday information.

The method we are proposing here is customized to fit our needs with the existing e-learning platform based on Moodle.

These needs include regular security updates from Moodle, intensive use of the calendar feature to schedule events and so on.

For these reasons, the fact that the central core of Moodle remains unscathed is a very big plus. It means that we don’t need to be mindful of small hacks, and where they need to go, after each update. It also means that in case we need to migrate, or to integrate this solution into another Moodle instance, we can do it as simply as installing a new plug-in. And the fact that the decision block only runs on the users’ machines and that it only calls the server in very specific conditions means that the server is not “bothered” with requests that can overrun it, even if most of the almost 5000 existing users are logged in at the same time.

Adding 20 lines of code into the template header, out of which maybe 5 are run on each Moodle page, doesn’t seem like an unfair trade.

V. FUTURE PERSPECTIVES

As stated by Graham [20], “the dialectic between the imaginations of supply side educationalists and technologists on one hand, and the desires, beliefs and aspirations of potential learners on the other”, meaning that we can develop, adapt and integrate a variety of tools, but only students can offer a real measure of their usefulness. Thus arises the need to have student evaluate the module in the near future.

As for future developments, we are considering publishing the information contained in microformats in RDFa (Resource Description Framework in attributes [21]), using existing vocabularies in their creation. A detailed comparison of microformats versus RDFa from the standpoint of applications for educational needs and also technological and semantic properties is presented in [22]. The main advantage of the RDFa publishing method is that it allows software agents to merge data from multiple sources, provided that they are described using the same vocabularies. By means of such queries, simultaneously run on multiple data sources, additional information could be obtained, information that otherwise, through simple source queries, would not be available. In order to reach this goal, the metadata could be converted into the RDF format, either by using scrapers / converters that extract information from microformats, or by using a tool that directly converts the information found in the database (D2RQ Platform – “system for accessing relational databases as virtual, read-only RDF graphs” [23]).

We also intend to release the final microformats plug-in into the Moodle community, once we are content with the functionality and the robustness of the code.

REFERENCES

[1] T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web," in *Scientific American* New York, 2001.

[2] R. Khare and T. Çelik, "Microformats: a pragmatic path to the semantic web," in *Proceedings of the 15th international conference on World Wide Web*, Edinburgh, Scotland, 2006, pp. 865 - 866

[3] J. Hebeler, M. Fisher, R. Blace, and A. Perez-Lopez, *Semantic Web Programming*: Wiley Publishing, Inc., 2009.

[4] E. Lewis, *Microformats Made Simple*. Berkeley, 2009.

[5] T. Çelik, "Microformats, Building Blocks, and You," 2007.

[6] Moodle.org, "Moodle Statistics." vol. 2012.

[7] I. Ermalai, M. Mocofan, M. Onita, and R. Vasiu, "Adding Semantics to Online Learning Environments," in *5th International Symposium on Applied Computational Intelligence and Informatics – SACI2009*, Timisoara, Romania, 2009, pp. 569-573.

[8] B. Dragulescu, I. Ermalai, M. Bucos, and M. Mocofan, "Using hCard and vCard for improving usability in Moodle," in *6th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI 2011)*, Timisoara, Romania, 2011, pp. 463-476.

[9] S. Lukichev, I.-M. Diaconescu, and A. Giurca, "Empowering Moodle with Rules and Semantics," in *SFSW*, 2007.

[10] O. Mustapasa, D. Karahoca, A. Karahoca, A. Yücel, and H. Uzunboylu, "Implementation of Semantic Web Mining on E-Learning," *Procedia - Social and Behavioral Sciences*, vol. 2, pp. 5820-5823, 2010.

[11] S. Leone, "Characterisation of a Personal Learning Environment as a lifelong learning tool," Università Politecnica delle Marche, Extended summary 2011.

[12] R. R. d. Oliveira, "Use of Twitter and Semantic Resource Recovery in the Educational Context," in *2012 IEEE 21st International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises*, Toulouse, Cedex 04, France, 2012, pp. 468-473.

[13] A. Rodríguez-Rodríguez, L. Cruz-García, C. Hernández-Guerra, and F. Quintana-Domínguez, "How Semantics can Improve e-Learning Tools," in *IADIS International Conference WWW/Internet*, MURCIA, SPAIN, 2006, pp. 364-366.

[14] M. Hendrix, A. Protopsaltis, I. Dunwell, S. de Freitas, P. Petridis, S. Arnab, N. Dovrolis, E. Kaldoudi, D. Taibi, and E. Mitsopoulou, "Technical Evaluation of The mEducator 3.0 Linked Data-based Environment for Sharing Medical Educational Resources," 2011.

[15] C. Bratsas, P. Bamidis, A. Dimou, I. Antoniou, and L. Ioannidis, "Semantic CMS and Wikis as Platforms for Linked Learning," in *World Wide Web* Lyon, France, 2012.

[16] P. D. Bamidis, S. T. Konstantinidis, C. Bratsas, and M. S. Iyengar, "Federating learning management systems for medical education: A persuasive technologies perspective," in *Computer-Based Medical Systems (CBMS), 2011 24th International Symposium on*, 2011, pp. 1-6.

[17] I. Hickson, "HTML Microdata - W3C Working Draft 29 March 2012," W3C, 2012.

[18] Tomas Steiner, Raphael Troncy, and Michael Hausenblas, "How Google is using Linked Data Today and Vision For Tomorrow," in *Linked Data in the Future Internet*, Ghent, Belgium, 2010.

[19] K. Goel, R. V. Guha, and O. Hansson, "Introducing Rich Snippets," G. W. C. Blog, Ed., 2009.

[20] G. Graham, "E-learning: a philosophical enquiry," *Education + Training*, vol. 46, pp. 308 - 314, 2004.

[21] W. C. W. Group, "RDFa 1.1 Primer Rich Structured Data Markup for Web Documents," 2012.

[22] V. Tomberg and M. Laanpere, "RDFa versus Microformats: Exploring the Potential for Semantic Interoperability of Mash-up Personal Learning Environments," in *Second International Workshop on Mashup Personal Learning Environments (MUPPLE09)*, Nice, France, 2009.

[23] C. Bizer, "D2RQ - Accessing Relational Databases as Virtual RDF Graphs."