

Automatic Generation of Mindmaps in Courses Implemented on Moodle Platform

Barbara Dębska, *bjdebska@prz.edu.pl*
Rzeszów University of Technology, Rzeszów

Łukasz Sanokowski, *l.sanokowski@pwsz.krosno.pl*
Higher State Vocational School, Krosno

Abstract

Among functions offered by Moodle, the most popular e-learning platform, there are not any algorithms which enable users to present graphically relations between items in the course and, therefore, systematize knowledge gained by students. This paper describes an application which can generate mindmaps based on the structure and hierarchy of educational materials included in a Moodle course. Mindmaps use images, associations and mnemonics to appeal to students imagination and, consequently, facilitate learning.

Keywords

mind maps, blended learning, didactics, Moodle

INTRODUCTION

In education, the content of a course is traditionally divided into sections called topics. Students and teachers work on each topic separately and do not go back to any item which has been covered and marked. However, knowledge is not comprised of distinct topics, but it is a living, variable structure whose elements are connected in a number of ways, such as inferences, reflection, point of view, associations, etc. As a result, it can be concluded that acquiring knowledge and developing skills means mostly discovering connections between concepts defined during the process of studying various areas of knowledge, rather than understanding descriptions of individual concepts.

One of the techniques used in creative education is drawing mindmaps which enable users to make a comprehensive analysis of the content of the course. The bases of this method were created by Tony Buzan (1990) who states that a mindmap is a diagram which includes the most important elements of a particular area connected with each other by means of associations. According to Buzan, the main topic (the root) which has to be studied, has to be placed in the middle of a mindmap. This central element is surrounded by other elements, called child nodes and linked to them with lines. Further elements associated with child nodes can be added to them, until all essential information is included in the mindmap. Apart from the theoretical bases of mindmapping, Tony Buzan has made some useful suggestions that enhance memorization. It is essential to bear in mind that people tend to remember things that occur at the beginning and at the end of a lesson; information which is unusual or unique is retained more easily than obvious facts; repetition and revision of previously acquired knowledge is of great importance.

Despite the fact that Tony Buzan emphasised the significance of mindmaps drawn by hand, he also worked on software called iMindMap, according to Pro-

jehtsukces webpage (2013), which makes it possible to apply a computer to generate mindmaps. Such mindmaps can be exported to commonly used files such as PDF, JPG, PNG as well as text files and used in presentations, lectures or publications. iMindmap is a commercial product, but there are some free tools, distributed as OpenSource software, which can be used to generate mindmaps, for example:

- MindMeister – the easiest tool to create mindmaps,
- XMind – a free application with an easy and intuitive interface which enables its users to move quickly through the mindmap using keyboard or generate a map in the form of an outline,
- FreeMind – software whose elements can easily be adjusted to users' needs (e.g. icons, colours, grouping, arrows). Mindmaps generated by this software can be exported to many popular formats (e.g. HTML, PDF, PNG). Another feature of this application is the ability to use mindmaps in e-learning as there is a special filter which enables users to display them on Moodle platform.

We decided to use FreeMind application while working on our project because mindmapping is regarded as one of the most efficient ways of acquiring knowledge during self-education. Research of Mikołajczyk (2005) has shown that knowledge gained during self-study is retained in memory longer and can be used more efficiently.

EXISTING MINDMAP TOOLS

Nowadays, there are several tools which can be used in education to create and explore mindmaps. Some of these tools are commercial, and some are free - distributed as OpenSource software. In this paper, the focus is on the latter, as free software is used in Higher State Vocational School in Krosno as often as it is possible. When choosing mindmapping software, the emphasis is put on its ability to use mindmap features inside Moodle, on which e-learning system called "Portal of virtual education e-Student" in HSVS in Krosno is based. As a result of conducted research, two types of applications, which fulfil the conditions mentioned above, have been found:

- Mindmap Module
- Freemind Filter.

Mindmap Module is an application formed as a Moodle module, which makes the following features available (see http://docs.moodle.org/23/en/Mindmap_module):

- creating and sharing maps online,
- publishing maps,
- exporting maps to files: XML, PDF or Freemind file format,
- editing mindmaps offline, using Gears or Adobe Air.

Freemind Filter is a Moodle filter type plugin, based on Freemind Flash Browser library (http://freemind.sourceforge.net/wiki/index.php/Flash_browser), which makes displaying Freemind maps files inside Moodle course pages possible. Such map files must be manually created using Freemind application, saved on computer hard disk drive and uploaded to a server. In conclusion, viewing is the only way of using mindmaps in Moodle through Freemind Filter, what is confirmed on webpage http://docs.moodle.org/20/en/FreeMind_filter.

Despite the obvious advantages associated with the concept of mindmapping, both of the described solutions have similar constraints. To be precise, maps must be created by hand (online or using offline application), which is a laborious and time-consuming process. It would be very difficult to maintain an up-to-date mindmap with the whole content of the course, as any change in the course would make information stored in it obsolete. Such a map would contain data common to all students, without any personalized information.

MINDMAP GENERATOR

This paper describes an application we designed and created, called "MindMap Generator". It is formed as a Moodle block, based on Freemind Flash Browser libraries and extends its functionality and enables automatic creation of mindmaps containing the whole content of the course. Such maps can be used to navigate through the course in an innovative way. What is more, the maps are always up-to-date as the application retrieves student's and course's data during the loading of the page containing the map. The application has been designed to present as much information as possible visually, using colours, font size and style, arrows, icons and so on. Each student uses a different map, because maps contain personalized information, which will be described in detail in the following chapters. The structure of a generated mindmap is presented in Figure 1:

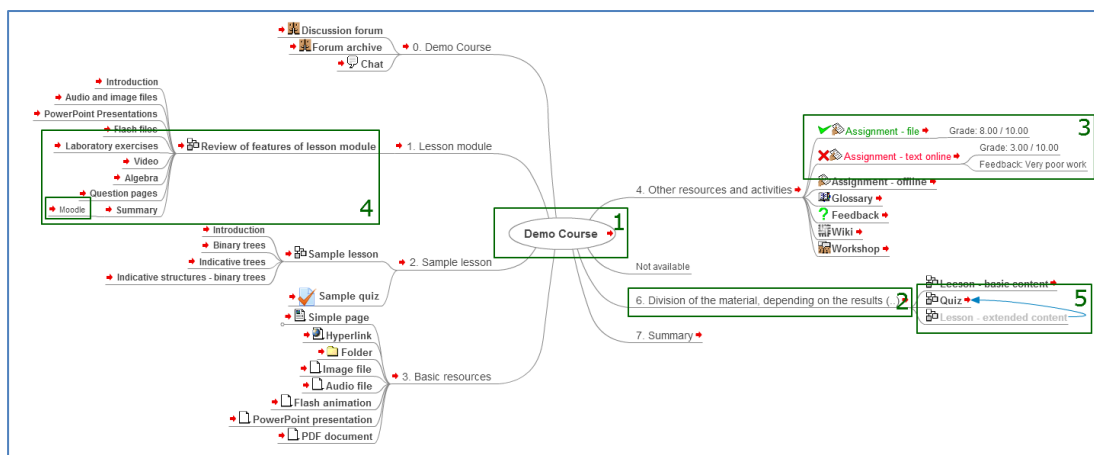


Figure 1: Mindmap containing course content

For comparison, the Figure 2 shows the same course in ordinary Moodle topics format:

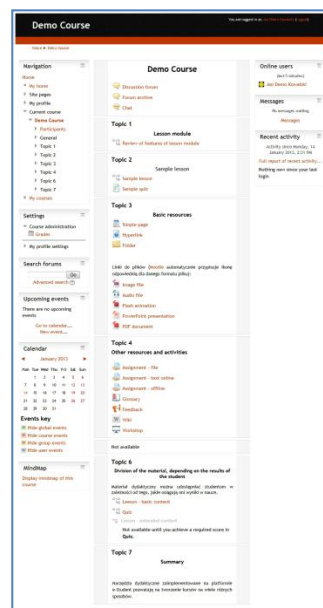


Figure 2: View of course in topics format

In this section, elements of the map will be described in detail, starting from the root which is a node which contains the full name of the course (Figure 3).

Child nodes of the root contain names of sections. If these names are too long, they are shortened to avoid wasting space on the map:



Figure 3: Root node

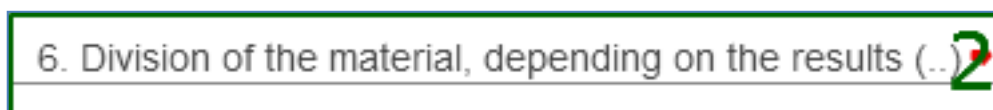


Figure 4: Section node

Child nodes of section nodes are activities and resources. Just by looking at this type of a node a student can see if he performed well in the module (if a particular module is gradable). If the font used to display particular activity name is green, it means that the grade he got is higher than or equal to the minimum grade defined by the teacher. Similarly, if the grade is lower, the font colour is red. What is more, next to the module's name an icon is displayed – a green 'thick' or a red 'X' respectively. If the minimum grade is not defined, the application will assume that the minimum grade is 50% out of the maximum grade available. As we can see in Figure 5, activity nodes can have child nodes, such as:

- a node with the amount of points achieved in activity, in relation to the maximum amount of points possible to gain,
- a node with teacher's feedback.

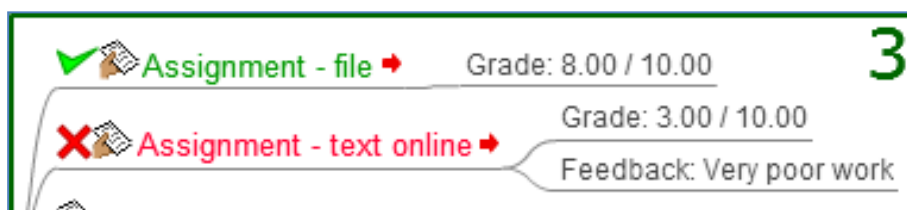


Figure 5: Nodes representing modules passed and failed with grades and feedback

In a lesson module, apart from grade and feedback nodes mentioned above, there can be additional child nodes, representing every page from the lesson's "left menu". These nodes are visible on the map only when the left menu is enabled and when the particular user has capability to see this menu.

What is more, if there is a glossary in the course, and if in that glossary a parameter called "Automatically link glossary entries" is enabled, each auto-linked glossary entry, which appears in the content of the course, is displayed as a child node of a particular activity node.

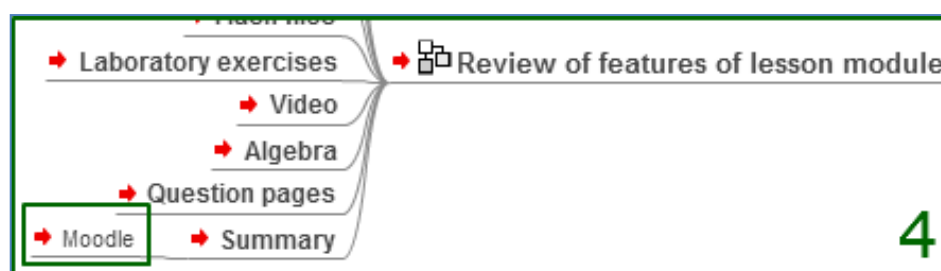


Figure 6: Lesson node with key pages and glossary entry in one of pages

FEATURES COMMON TO ALL NODES

Some of features are common to all nodes, namely:

- All nodes, which can be seen by a user, are 'clickable' which is indicated by an arrow icon next to their names. The arrow is a hyperlink leading directly to a particular content, which is displayed in a new browser tab. For example, a student can use a hyperlink to go directly to a particular page in the lesson or a specific section of the course.
- All activities and resources have appropriate Moodle icons next to their names.
- If a particular element is hidden, but the user has capability to see it, its name is displayed in grey font.

FEATURES NOT AVAILABLE IN MOODLE SO FAR

The most important and innovative feature of Mindmap Generator is displaying the whole content and structure of the course as a mindmap. Furthermore, the application offers additional features, not available in Moodle or in contributed modules so far, such as:

- Depicting conditional activity dependencies. It means that the application checks in the database, if there are any activities on which a particular activity depends. After that, arrows which start at the access-restricted activity are drawn on the map. The arrowheads indicate activities on which dependency condition is set. Dependencies between lesson modules, which are a legacy from earlier Moodle versions, are also recognized and visualized.



Figure 7: Conditional activity dependency

- Depicting elements that have not been seen by a user in bold font, even particular pages of lessons. Therefore, a user can check how much content of the course he has seen, and how much has remained, at a glance.
- Colouring modules which were failed or passed. It allows the user to get a quick review of his performance in the course (this feature was described in detail in previous chapter).
- A search engine, providing fast browsing through section names, module names, lesson key pages titles and glossary entries.

DEVELOPMENT PLANS

This chapter describes features, which are already been designed and will be developed in future. Namely, two innovative methods of using mindmaps in education have been designed. Both of them are based on a feature which allows users to download a map file to their computer and open it with Freemind application.

1. The first option will enable teachers to configure the maximum number of node levels in maps created by Mindmap Generator. For example, if the teacher set the maximum number of node levels to 2, the map would contain only the root node with the name of the course and nodes with section names. Further child nodes such as modules or resources nodes and their child nodes, would not be visible on the map. The task for the students would be to download the map file and complete the missing elements of the map by hand, using Freemind application. The completed maps could be assessed by the teacher afterwards. The second possibility for the teacher is to download the map file and erase some of content. For instance, he could whiten some part of the module or resource names, remove or modify some elements, etc. The stu-

dents' task would be to complete the missing elements, or correct the errors. What is more, in all of the examples described in this chapter, the teacher might prepare lists of items, which could be used by students to complete the maps. Some of them could be correct, while others could be false.

2. The second feature which will be developed will automatically highlight elements which have to be passed and the deadline is in the near future. Such activities will be highlighted with yellow font colour and special exclamation mark icon if they fulfil both of the following conditions:

The first condition: the student hasn't made an attempt in the activity, or the grade he got for it is not high enough to pass. The application will assume that the minimum grade to pass is 50%, unless the teacher set a different value of this parameter in the gradebook.

The second condition: less than 20% of the total time is left for completing the assignment or passing the quiz. The application will compute the total time (it means time when the activity is available to the student) in activities which will have opening and closing dates configured.

If only the closing date is set (without the opening date parameter), it will not be possible to compute the total time during which the activity will be available. In such a case the activity will be highlighted if there is less than 7 days left until the deadline.

After implementing this feature, there will be possible quick distinction between activities, based on three colour schema: green, yellow and red. It is depicted in Table 1.

Table 1: Three colour schema

   Assignment - file	Passed activity
   Assignment - offline	Activity which was not passed yet, and is close to deadline
   Assignment - text online	Activity which was not passed

3. One more planned feature is a function similar to Moodle's "Switch role to..". Namely, the teacher will have possibility of viewing and downloading mindmaps as if he were a chosen student. It could be used by the teacher to display mindmaps from a particular student's perspective. This new feature, available in Mindmap Generator will enable teachers to quickly visually check the students' progress and performance in the course.
4. Once implementing the features in development plans, publishing the application in Moodle plugins database is intended. After submitting the application it will be tested by a specialist from Moodle HQ. Only after the approval by a Moodle employee, the application may be made available to other users, so this process gives a good measure about particular application quality.

The development of Mindmap Generator

During the coding process of Mindmap Generator, the emphasis was put on creating a fast, secure and compatible code. It has been achieved because guidelines which are described on webpage: <http://docs.moodle.org/dev/Coding> were strictly followed.

The existing functions from genuine Moodle library files were used as often as possible. New PHP functions or database queries were created only when necessary. This approach also has resulted in the acceleration of application's performance, because it entails using Moodle caching system.

Initially, the core of the application was created in Moodle 1.9 version and its features worked properly. When Moodle 2.0 version emerged, it was correctly foreseen that the new version would replace Moodle 1.9 and the process of the development of the legacy version would stop. A decision was made to immediately modify the application in order to achieve compatibility with the new version of API. What is more, it was assumed that Mindmap Generator would be adapted to new Moodle releases as they emerge in the future.

THE CONCLUSIONS DRAWN DURING THE PROCESS OF DEVELOPING ADDITIONAL MOODLE MODULES

Our team has experience in the field of designing and developing new applications and features in Moodle system. For example we successfully created and published an application called Timestat Block, which can be downloaded from webpage: <http://moodle.org/mod/data/view.php?d=13&rid=4200>.

It has been observed that the process of Moodle development is relatively fast and meaningful, which is important for Moodle users. For example, they benefit from the fact that features and modules are created or improved, and bugs and errors are fixed. However, changes in Moodle Application Programming Interface entail impediments for contributed code developers. They are forced to do additional work to improve their applications so that they are compatible with current Moodle version. It seems that the migration from Moodle 1.9 to 2.0 was more difficult than other, because significant changes in Moodle 2 release. For example, the file storage mechanism was completely rewritten. In our opinion, this change was unnecessary and resulted in more difficulties than benefits. What is more, most of other core functions were modified, for instance output API, data access API and so on. A similar situation occurred after publishing following 2.x releases. Fortunately, in these cases the changes were not as burdensome.

Interestingly, the biggest difficulty for the developer was the lack of up-to-date documentation. The developer documentation available was not updated for a long time after releasing new Moodle versions. It was a frequent situation that all pages for particular API documentation were marked as obsolete and there was no reference to any proper and current data. As a result, it was necessary to analyze the existing Moodle source code, in accordance with the principles of reverse engineering, in order to use of generic functions correctly, as it was the only source of valid information. Users who want to create new activities or modify existing code are discouraged and frustrated by this inconvenience.

CONCLUSION

To sum up, all techniques enhancing learning ought to be implemented in the process of e-learning on educational platforms. Teachers can use mindmaps to check whether all required topics are included in materials students receive. Information and associations in mindmaps presenting the content of the course and generated for a particular student are clear and easy to understand. Such mindmaps can also be developed by a student and include his personal associations, thus stimulating his imagination with images, symbols and keywords. Moreover, the application can be used in e-learning to monitor students' progress. If a student is able to reconstruct an element of a mindmap deleted or modified by his teacher, it means that he acquired a satisfactory level of knowledge and is able to recognize associations correctly.

This application can also generate a map which presents not only the content of the course, but also students' involvement in the process of self-education. Icons, colours, arrows and other graphic elements make mindmaps dynamic and cause changes in their structure over time. Keywords on mindmaps are linked to a glossary which helps students to memorize their definitions due to the fact that they can

refer to it as often as they want to. Mindmaps can also notify students of a close deadline for an assignment.

The application generating personalized mindmaps for individual students is a great development of existing features of Moodle platform.

REFERENCES

Buzan T., Buzan B. (1990) *The Mind Map Book*, Thorsons Publication

Freemind Flash Browser description, retrieved in May 2010 from:

http://freemind.sourceforge.net/wiki/index.php/Flash_browser

Freemind Filter documentation, retrieved in May 2010 from Moodle documentation:

http://docs.moodle.org/20/en/FreeMind_filter

Moodle coding guidelines, retrieved in 2012 from: <http://docs.moodle.org/dev/Coding>

Mindmap Module documentation, retrieved in December 2012 from Moodle documentation: http://docs.moodle.org/23/en/Mindmap_module.

Projektsukces, retrieved in December 2013 from: <http://www.projektsukces.pl/mapy-mysli-programy.html>

Timestat Block available to download from Moodle Modules and Plugins database: <http://moodle.org/mod/data/view.php?d=13&rid=4200>

Biographies



Barbara Dębska is an associate professor at Rzeszów University of Technology, and Head of the Department of Computer Chemistry. Her main research interests are chemometrics, expert system technologies and designing decision support systems for chemistry and the chemical industry. Another field of her interest is e-learning methods in teaching and learning in higher education.



Łukasz Sanokowski is an IT and e-learning specialist in Higher State Vocational School in Krosno. His main interest is Moodle, and network and Linux security.

Copyright

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 Unported License. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/3.0/>