Statement

I hereby declare that this diploma thesis is completely my own work and that I used only the cited sources.

Pilsen, 19. 5. 2011, Michal Campr,

Acknowledgements

I thank the head of my thesis Ing. Martin Zíma, Phd. for his guidance during my work on this diploma thesis. I also thank my family, my relatives and all my friends, who have helped me during my studies.

This research was supported by the EU and the Ministry of Education of the Czech Republic, project No. CZ.1.07/1.2.00/08.0021

Abstract

This diploma thesis deals with creation and maintanance of a web application for project ARET (Automatic Reading of Educational Texts), which is being developed on the Department of Cybernetics, University of West Bohemia. The first goal was to develop an administration interface for creating and editing educational text mainly for students of primary schools with some form of visual handicap. These educational texts are primarily focused on mathematics and physics, so a specialized tool for managing mathematical formulas was developed. The second goal was the development of a public interface for presentation of these texts. The presentation is required to be not only visual, but it should also use other means of communication, such as audio. More specifically with the use of text to speech synthesis, so that the educational text can be read aloud to the students (including the mathematical formulas).

Contents

List of Figures

Chapter 1

Introduction

1.1 Project ARET

The purpose of this project is the innovation and development of learning methods, especially the home preparation for students. This project is specifically aimed for students of primary schools with some form of visual handicap. These students are also often afflicted by some form of learning disability and have problems with home preparations and with using textual materials like textbooks or notebooks with their own notes, which are often of very low quality and thus not usable for learning. In general, working with extensive textual materials is a very big problem for these students. Even with the use of specialized tools, like magnifying glass or books in Braille, they have to pay extra attention to reading, so if the text is not read correctly, the information is misunderstood and it leads to demotivation and decreased willingness for education. This problem is even larger, when the students are confronted with mathematical formulas and symbols, which are often very confusing.

The outcome of this project is supposed to be a new form of learning tools and its real application in primary schools. Specially prepared learning texts are to be automatically read to students via a web application which uses automated speech synthesis. This will lead to more effective learning and the students will be able to fully focus on the content, rather than on reading the text. The control of the automatic reading must be implemented by a simple interface, so that the students have no difficulty using it. The students are nowadays quite familiar with working with computers, so controlling the created interface should not be a problem.

The created application is expected to be used in more schools than it is specified in the projects requirements. The created textbooks, and the technology used for their reading, can be used in other schools for visually impaired students or in schools for students with learning disabilities. The fact, that the project focuses on reading mathematical and physical texts, should guarantee that the system is universal and can be used for creating textbooks with other topics. As was mentioned, the first topics are supposed to focus on math and physics for second grade of elementary schools. Two types of learning tools are to be created:

1. Textbooks focused on specific topics - their content will consist of definitions and explanations of mathematical or physical terms, descriptions of experiments and examples of solving arithmetical problems.

2. Collections of arithmetic problems - their content will consist of examples for home preparation with necessary instructions for solving them. These examples will be of various difficulties.

The primary partner of this project is the Elementary school and Kindergarten for visually impaired students in Pilsen. Specialized teachers of this school will have the responsibility for creating the contents of the learning materials and their use in education. Improvement of home preparations of students with learning disabilities will lead to easier understanding of given topics and thus to their successful transition to a secondary school.

1.2 Problem description

As a part of the project *ARET*, this diploma thesis is dealing with the development of the software for creation, management and presentation of specialized textbooks for visually impaired students. This software is supposed to be in a form of a web application which can be accessed online by the students as well as by the teachers who are responsible for creation of the educational texts. The web application therefore consists of two parts - frontend (i.e. the public interface for presentation of the texts to the students) and backend (i.e. the administration of the database with educational texts).

The public interface (frontend) should consider all recommendations related to web accessibility for blind or otherwise visully impaired users. It means that the interface should be easy to use and understandable as well as correctly structured. In addition, the frontend is to be provided by a specialized audio player, which will use the automated speech synthesis (provided by the Department of Cybernetics, University of West Bohemia), to read the contents of the eduacational texts to the user. This player should also be simple and provide an interface, which would be easy to use.

The administration of the web application (backend) and its various parts are also required to be simple and understandable for non-frequent internet users. This means that every information and tool should be easily accessible and available in a matter of a few mouse clicks. The administration will also be equipped with specialized tools for inserting mathematical and physical formulas and other special semantic elements into the HTML code, which will be displayed and read in the public interface.

1.3 Contribution of this diploma thesis for the project

The actual contribution of this diploma thesis to the project in general is the administration of the whole application on the server, the responsibility for the correct functionality and adjusting the application in order to resolve the problems and requirements of the final users.

More specifficaly, the work on the application included creation of several plugins for the *CMS (Content Management System)* provided by the Department of Cybernetics and linking the existing frontend a backend with web services (also provided by the Department of Cybernetics), which provide various conversions connected to the speech synthesis (e.g.

MathML to text or TEXto image). In addition to that, creating several plugins for the *TinyMCE* editor in order for it to be able to insert mathematical or physical formulas into the content. In addition to that, the frontend was enhanced with an audio player connected to the web services, which provide the text to speech synthesis.

All of these features will be thoroughly discussed in chapters 5 (*System architecture*) and 6 (*System implementation*).

1.4 Document structure

The whole document is divided into two main parts. The first part contains the essential theoretical facts, which are required to be known in order to understand the second (implementation or practial) part.

The theoretical part is composed of three chapters and their several sections:

- Chapter 2 - Web accessibility - describes the requirements of blind or visualy impaired computer users and what can be done to ease the use of the web for such users.
- Chapter 3 - Web applications in general - describes the means of creating web applications, mainly the available technologies and which one is most suitable for the project ARET.
- Chapter 4 - Web content management - contains the description of several applications, which can be used for managing the contents of a web page.

The implementation (or practical) part consists of two chapters:

- Chapter 5 - System architecture - describes in theory how the whole application works and what logical parts it consists of.
- Chapter 6 - System implementation - describes in detail the outcome of this diploma thesis, what was created and how it works in cooperation with other logical parts of the application.

The last chapter (7) contains the evaluation of the final application in use, possible future upgrades and the overall result of this diploma thesis.

This document also includes an appendix, which contains the list of abbreviations.

Chapter 2

Web accessibility

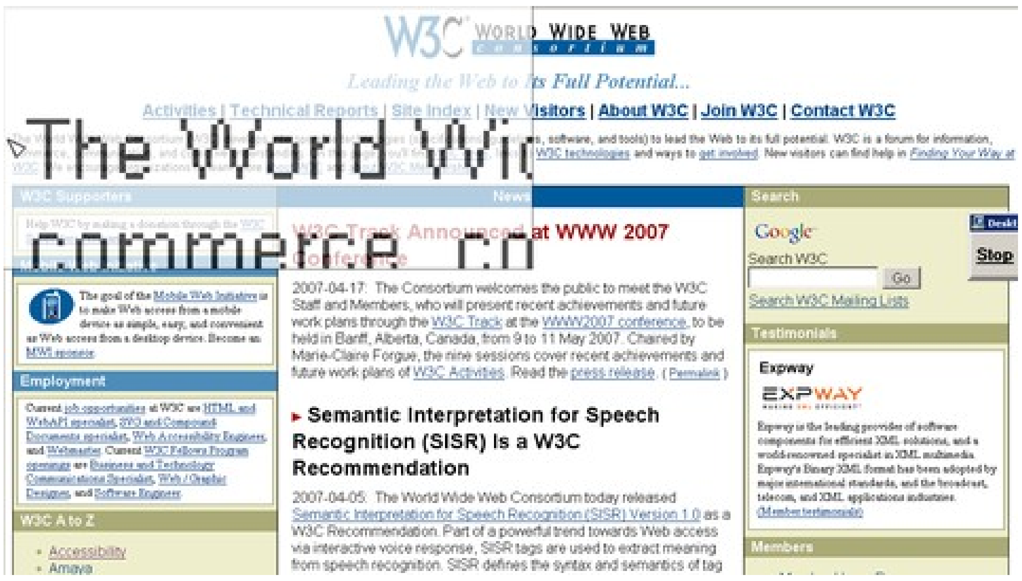
This particular chapter will discuss the subject of internet accessibility for visually impaired users, i.e. users who are either blind or suffer from some form of visual disorder. Visual disorders can take many forms (depending on the cause of the disorder) such as some forms of color-blindness, nearsightedness or farsightedness. There are also some not so well known forms, such as the inability to localize objects or follow objects in motion. Since this diploma thesis deals with a specific project for a specific group of visually impaired students, the first section of this chapter will discuss only some forms of visual disorders and requirements necessary for those users to interact with computers and to work with the internet. The second section will discuss what is necessary for reading the internet pages to the user and what applications are already available. In the last section of this chapter will be summarized available applications for reading mathematical formulas for visually impaired computer users.

2.1 Requirements of visually impaired users

Because of the large variety of visual disorders, there can be many types of tools, which are able to lower the negative effects of the particular disorder or even completely eliminate them. This means that the user can use these tools to access information through his computer as easily as any user with normal vision. This can be accomplished either by enhancing the visual information or using other senses to receive the information. These tools can therefore be divided into three categories depending on the sense, which is used to deliver the information: visual, audio and tactile tools. Besides receiving information from the computer, some users also have to use specialized tools for the input (i.e. writing text). These can be for example specialized keyboards or another additional tool like the Braille display.

The tools, which are aimed at enhancing the visual information, are not intended to be used by the blind users or by users with very poor vision. These tools are meant to enlarge the textual information on the screen or somehow modify the content to be easily readable and recognizable. For example, changing the colors to gain more contrast between the objects on the screen or between the text and the background. These tools are commonly known as a *software magnifying glass*. Though magnifying software is already a part of some operating systems, some additional software is required for most of the visually impaired users. These applications offer some additional functionality, like the mentioned contrast changing or color adjusting. There are many applications for magnifying the content on the screen, some are free and can be downloaded (*DesktopZoom*, *iZoom* [1]) but the more advanced applications have to be payed for (*The Magnifier* or *MAGic Screen Magnification Software* [2]). Which application is chosen by the user depends solely on his preferences and his actual disability. An example of such a software is in the figure 2.1.

Another way of receiving information from a computer can be using an audio output. This type of software is called a *screen reader*. This application attempts to identify all relevant information on the screen and then interpret them either with the use of a text-to-speech synthesis as a sound (voice) or sending the information to a Braille output device (or both sound and on Braille output). There are many companies which are creating screen readers. There are some applications which can be freely downloaded, some are even open source. Examples of open source applications are: *NVDA - Non Visual Desktop Access* which makes use of the speech synthesizer engines that come with Windows XP and Vista, *eSpeak* or *Thunder Speech Reader - the Talking Software*. However, large majority of the released


W3C WORLD WIDE WEB consortium

Leading the Web to Its Full Potential...

Activities | Technical Reports | Site Index | New Visitors | About W3C | Join W3C | Contact W3C

The World Wid

commerce, co

..., software, and tools) to lead the Web to its full potential. W3C is a forum for information, ... W3C technologies and ways to get involved. New visitors can find help in Finding Your Way at W3C ...

W3C Supporters | News | Search

Help W3C by making a donation through the W3C ...

W3C Track Announced at WWW 2007 Conference

2007-04-17: The Consortium welcomes the public to meet the W3C Staff and Members, who will present recent achievements and future work plans through the W3C Track at the WWW2007 conference, to be held in Banff, Alberta, Canada, from 9 to 11 May 2007. Chaired by Marie-Claire Forgue, the nine sessions cover recent achievements and future work plans of W3C Activities. Read the press release. (Permalink)

Semantic Interpretation for Speech Recognition (SISR) Is a W3C Recommendation

2007-04-05: The World Wide Web Consortium today released Semantic Interpretation for Speech Recognition (SISR) Version 1.0 as a W3C Recommendation. Part of a powerful trend towards Web access via interactive voice response, SISR tags are used to extract meaning from speech recognition. SISR defines the syntax and semantics of tag

Mobile Web Initiative

The goal of the Mobile Web Initiative is to make Web access from a mobile device as simple, easy, and convenient as Web access from a desktop device. Become an MWI sponsor.

Employment

Current job opportunities at W3C are HTML and WebAPI specialist, SVG and Compound Documents specialist, Web Accessibility Engineer, and Webmaster. Current W3C Fellows Program openings are Business and Technology Communications Specialist, Web / Graphic Designer, and Software Engineer.

W3C A to Z

- Accessibility
- Amaya

Google

Search W3C

Go

Search W3C Mailing Lists

Testimonials

Expway

EXPWAY

Expway is the leading provider of software components for efficient XML solutions, and a world-renowned specialist in XML multimedia. Expway's Binary XML format has been adopted by major international standards, and the broadcast, telecom, and XML applications industries. (Member testimonials)

Members

Deskt

Stop

Figure 2.1: Screen magnifier (left upper part of screen is magnified) [1]

software is quite expensive (*ZoomText* or *JAWS - Job Access With Speech*) [2] and can be worth as much as £800.

Many of the screen readers can also be used to transform the content of the screen into a textual information and send it to a specialized output device which is able to display it in Braille alphabet (figure 2.2). Each Braille character (cell) is made up of six dot positions arranged in a rectangle containing two columns of three dots. A dot may be raised at any of the six positions to form 64 possible combinations. Any possible combination can be used to represent a single letter or a special sign like *comma* or *semicolon*. [2] The devices for displaying Braille script are called *Refreshable Braille displays*. On this display, the Braille characters can be interactively displayed to the user depending on the content of the screen. Some more advanced displays come with two additional dots on the bottom of every character, which can show the position of the cursor in the text. They can also be used for advanced maths work and for computer coding. An example of such a display is in the figure 2.2. These displays depending on their complexity and accessories can cost as much as $10,000, so their accessibility is limited.

In addition to that, managing mathematical formulas through the Braille display can

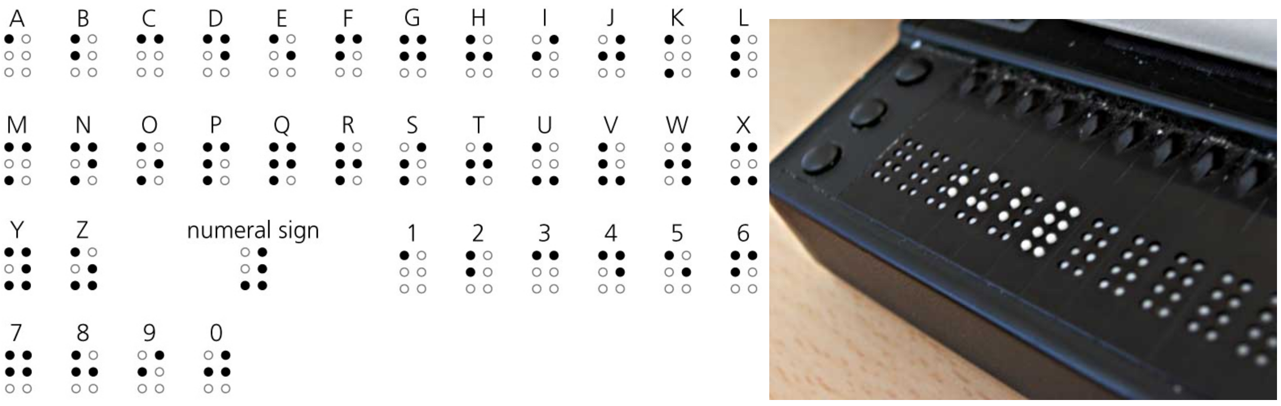


Figure 2.2: Braille alphabet, Braille 8-dot display [2]

be also problematic. There is one well known project which deals with this problem. The project *Lambda* aims at solving the problem of mathematics text management by blind users of secondary schools and universities, as well as that of fruition of science texts, in digital formats and through Braille print.[3]

2.2 Reading of web pages

Reading of internet pages can be thought of as a part of the screen reader problem. The goal is to search for all relevant information, which are displayed in the web browser window, and interpret them as a sound, i.e. read them aloud to the user. There are several applications, which are trying to deal with reading the web pages without the use of a screen reader, which is already installed on the computer. These applications (e.g. *Spoken-Web* [4] or *BrowseAloud* [5]) can be downloaded and installed into the browser as a plugin or as a standalone application. The fact, that all of these applications need to be installed, can be viewed as a disadvantage and introduce some new problems, like system (or browser) incompatibility and requirements issues.

The application, which was developed within the project ARET, aims to introduce a new way of reading web pages and include some additional functionality. The final web application is able to select relevant content of the particular page and with the use of standard internet technologies (*JavaScript* and *Flash* or *HTML5*) read them to the user. The need to install any extra software is therefore eliminated. In addition to that, the text-to-speech synthesis, which is used for the application, is upgraded with additional modules for reading mathematical and physical formulas. The final application can therefore be used as an online educational tool for visually impaired students.

2.3 Reading of mathematical formulas

Reading of mathematical or physical formulas can be a difficult task for the text-to-speech synthesis. The formula's structure and complexity is very variable and the meaning of symbols can be also difficult to recognize. A few applications exist (e.g. *MathPlayer* [6]) that try to deal with the problem of math-to-speech synthesis, however a similar problem like with the reading of web pages is present here: these applications have to be either installed as a plugin to a specific web browser (mostly *Internet Explorer*) or be installed as a standalone application into the operating system. The mentioned application is also limited by language, focusing only on English language, which is not to be used within the project ARET. The developed application is able to generate a TEXand MathML representations of a given formula, which was created by a teacher in a specialized *WYSIWYG* (What You See Is What You Get) editor. These representations are then used for generating an image of the formula (TEXto image) and a textual description of it (MathML to text). The result is saved to the HTML code of the web page as an *IMG* tag with a given alternative textual description. This description is then read to the user when the page is displayed.

Chapter 3

Web applications in general

This chapter will describe some of the most often used technologies for creating web applications and their advantages and disadvantages. There will also be discussed, which one of these technologies was the most suitable for developing the web application within the project ARET.

3.1 Web application

A *web application* is a specialized application, which can be accessed over the internet via a web browser. This application can be looked at in two ways. The application core is stored on a server and it is called by the client (the web browser). Depending on which side we focus, the application can be divided into a client side and server side application.

When the client sends a request for a specific page (i.e. requesting an URL in a web browser), it travels through various network components until it reaches the requested server. The server then processes this request, generates some specific data, usually in the form of a *HTML* or *XHTML (Extensible HTML)* code, and sends the response back to the client. The client then diplays the data to the user.

3.1.1 Client side

On the client side, there are specific technolgies used for displaying and managing the content of the web page. The code of the web page, which was assembled on the server, consists of several parts. The HTML or XHTML code can also includes some *JavaScript* code, which can interact with the content in the browser. The JavaScript language also has two very popular extensions - *AJAX (Asynchronous JavaScript and XML)* and *jQuery* library, which are often used for further modifications of the page content and asynchronous communication with the server. Among the essential web technologies can also be included the *CSS (Cascading Style Sheets)*, which is used to separately define the visual aspects of the given web page. All of these technologies are used to display the desired information to the client but have no influence on how to get the particular information and where it is stored (i.e. the server from which the data originate).

3.1.2 Server side

On the server side, the application is required to gather the required data (depending on the request from the client), assemble the resulting code and send it to the client. There are, however, many technologies for creating the server side applications. There are many frameworks built upon different programming languages, which are adapted on creating web applications. Each language has its specific advantages and disadvantages, as do the frameworks. Some of them are more suited for developing large and robust commercial projects, others are well suited for quick development of smaller scale applications. The following sections will describe the most popular technologies, which are used for creating web applications, and will put them into the context of the project ARET and its needs and requirements.

Java

For running a java-based web application, a java-enabled server with installed *JVM (Java Virtual Machine)* is needed. The core of any java-based application is called a *Servlet*. A servlet is a protocol and platform independent component, which runs on the server and answers any requests that come from the clients. Despite the protocol independence, the servlet technology is nowadays used mainly with the HTTP protocol, so it is used for building entire web pages on the fly. Java servlets can do several things that are difficult or impossible with other applications. For example, servlets can directly communicate with the Web server. This simplifies operations that need to look up images and other data stored in standard places. Servlets can also share data among each other. They can also maintain information from request to request, simplifying things like maintaining sessions and caching of previous computations.

There is an additional technology, which can be used together with servlets: *JSP (Java Server Pages)*. It is a technology that can be used to mix regular, static HTML code with dynamically-generated HTML. JSP can create the two parts separately, i.e. insert dynamic HTML code, which can be created by a servlet, into the static HTML code.

Creating larger java-based applications can be made easier by using any of the available frameworks, like *Spring*, *Hibernate* or *Bento*. The most common, however, is the Spring framework. Despite being very popular, this framework, like the others, is quite complicated to learn and use, because there are many information, which the developer has to know, before he can start developing. There are also more choices to be made regarding other supporting technologies (e.g. which servlet container to use). This provides some inconveniences for quickly starting the development and for the server environment setup.

ASP.NET

Unlike the other technologies listed in this chapter, *ASP.NET (Active Server Pages)* is not a programming language, but it is a technology very often used for developing web applications. ASP.NET is a web application framework developed by Microsoft for building dynamic web applications and web services. It is the successor to Microsoft's *ASP (Active Server Pages)* technology. ASP.NET is built on the *CLR (Common Language Runtime)*, allowing development using any supported .NET language (C#, C++, J# etc.). This technology is a direct competitor of the JSP technology. It tries to concentrate on some problems, which can occur using the JSP technology and remove them, but it has some limitations concerning programming language and run-time environment.

Python

It is possible to use maybe any available programming language for creating web applications, but using Python on its own is a very difficult task. Because of this, there are many frameworks to use, for example *Django*, *Grok*, *Pylons* or *TurboGears*. One of the more popular frameworks is *Django*. Its primary goal is to ease the creation of complex, database-driven websites. Django emphasizes reusability of components, rapid development, and the principle of *DRY(Don't Repeat Yourself)*. It also provides an optional administrative CRUD (create, read, update and delete) interface that is generated dynamically and configured via admin models. [7]

Ruby

Ruby is a dynamic, general-purpose object-oriented programming language that combines syntax inspired by Perl. There is currently no specification of the Ruby language, so the original implementation is considered to be the *de facto* reference. As of 2010, there are a number of complete or upcoming alternative implementations, including *YARV*, *JRuby* or *Rubinius*. However, for developing web applications, there are also several frameworks for Ruby language, including *Merb*, *Camping* or *Wuby*. One of the most popular frameworks is *Ruby on Rails*. Ruby on Rails includes tools that make common development tasks easier, such as scaffolding that can automatically construct some of the models and views needed for a basic website. Also included are *WEBrick*, a simple Ruby web server that is distributed with Ruby, and *Rake*, a build system, distributed as a gem. Together with Ruby on Rails these tools provide a basic development environment. [8]

PHP

PHP is a scripting language originally designed for web development to produce dynamic web pages. For this purpose, PHP code is embedded into the HTML source code and interpreted by a web server with a PHP processor module. It also has evolved to include a command-line interface capability. PHP can be deployed on most web servers and as a standalone interpreter, on almost every operating system and platform for free.

The PHP source code is compiled on-the-fly to an internal format that can be executed by the PHP engine. In order to speed up execution time and not have to compile the PHP source code every time the webpage is accessed, PHP scripts can also be deployed in executable format using a PHP compiler. The compiled code can be further optimized to enhance the performance by reducing its size, merging redundant instructions and making other changes that can reduce the execution time. Another way of reducing compilation count is using a *opcode cache*. Opcode caches work by caching the compiled form of a PHP script (opcodes) in shared memory to avoid the repeated compiling of the code every time the script runs.

There are many frameworks, which are built with the use of PHP language. Among the more popular are *Zend*, *Code Igniter*, *Smarty* and *Symfony*. In the Czech Republic is also very popular framework *Nette*. For developing the application for the project ARET, the programming language PHP has been chosen and along with it, the framework Symfony, which will be described in the following section.

3.2 MVC architecture

All of the before-mentioned web application frameworks share a common feature concerning the general architecture of the web application. This architecture is called the *MVC(Model View Controller)* design pattern. In short, the MVC design pattern defines a way to organize the code according to its nature. This pattern separates the code into three layers (figure 3.1):

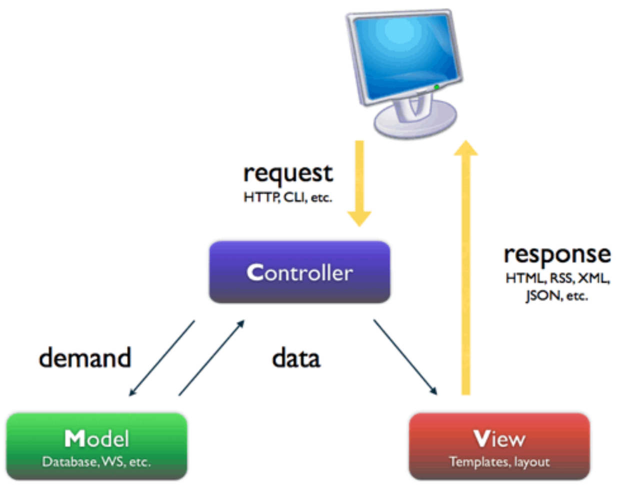


Figure 3.1: The MVC design pattern structure [9]

The Model layer defines the business logic (the database belongs to this layer). Symfony stores all the classes and files related to the Model in a separate directory.

The View is what the user interacts with (a template engine is part of this layer). In symfony, the View layer is mainly made of PHP templates. They are also stored in various directories.

The Controller is a piece of code that calls the Model to get some data that it passes to the View for rendering to the client. All requests that come from the client are managed by front controllers. These front controllers delegate the real work to actions. These actions are logically grouped into modules.

3.3 PHP + Symfony 1.4 + Doctrine 1.2

Symfony is a web application framework for PHP projects, which provides very quick development and maintenance of web applications. It aims to remove the repetitive coding tasks and enhance the control over the whole project. The very small number of prerequisites makes symfony easy to install on any configuration - only a Unix or Windows with a web server and PHP 5 installed are needed. It is compatible with almost every database system. In addition, it has a very small overhead, so the benefits of the framework don't come at the cost of an increase of hosting costs. Symfony is aimed at building robust applications in an enterprise context. This means that the developer has full control over the configuration: from the directory structure to the foreign libraries, almost everything can be customized. A very important feature of Symfony is an active open-source community. It is entirely free and published under the MIT license. Currently, a beta of Symfony2 is also available. [9]

Doctrine is a PHP *ORM (Object Relational Mapper)* for PHP 5.2.3+ that sits on top of a powerful PHP *DBAL (Database Abstraction Layer)*. One of its key features is the ability to optionally write database queries in an *OO (Object Oriented)* SQL-dialect called *DQL(Doctrine Query Language)* inspired by Hibernate's HQL. This provides developers with a powerful alternative to SQL that maintains a maximum of flexibility without requiring needless code duplication. [10]

The framework Symfony combined with the ORM Doctrine meets every essential requirements of the project ARET and provides many useful features for fast development. The most important aspect of symfony is a powerfull and easy-to-use plugin system, which allows integrating many of the available opensource plugins into the final system with a minimum effort.

A symfony plugin offers a way to package and distribute a subset of project files. Like a project, a plugin can contain classes, helpers, configuration, tasks, modules, schemas, and even web assets. [9]

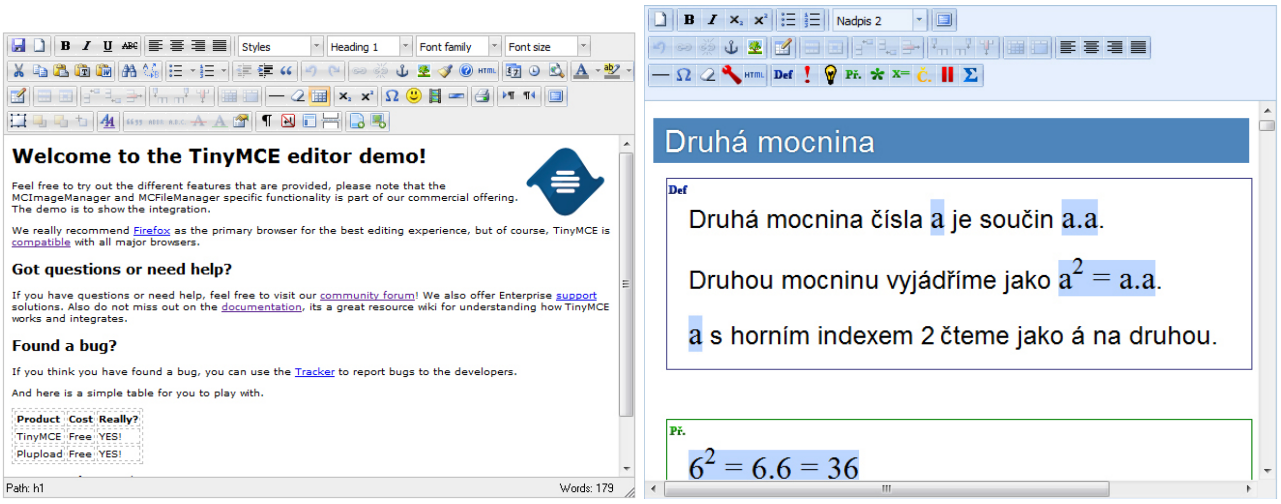
Since the plugins are very easy to create and update, it is possible to work on a specific functionality within one plugin and then upload it and include it in the final application. For example: develop a HTML content editor which contains a formula editor and then include it in an already complete administration interface.

Chapter 4

Web content management

Every web application (web page) is based on displaying some relevant information to the user based on his request. This information can be some automatically generated data, images, audio or video files or just textual information. They are not put there by the server, but by the administrator, who is a person responsible for the content. He is able to manage the content typically via some form of administrative tool for editing the content, accessing the database and updating the data. In the application, that is being developed within the project ARET, it is necessary to administer the HTML content of each topic, the database of images and the database of authorized users.

Besides HTML content, there is the necessity of creating and editing mathematical and physical formulas. A specialized tool for managing the formulas was developed. There are also more types of data (e.g. sound files or images of formulas) stored on the server, but they are being automatically generated according to the content inserted into the HTML document. These data do not need to be administered, so there has to be no additional tool for editing them. There are several tools for managing the HTML code and creating mathematical formulas in the environment of a web browser and some of the more popular will be discussed and described in the following sections of this chapter.



Welcome to the TinyMCE editor demo!

Feel free to try out the different features that are provided, please note that the MCImageManager and MCFileManager specific functionality is part of our commercial offering. The demo is to show the integration.

We really recommend Firefox as the primary browser for the best editing experience, but of course, TinyMCE is compatible with all major browsers.

Got questions or need help?

If you have questions or need help, feel free to visit our community forum! We also offer Enterprise support solutions. Also do not miss out on the documentation, its a great resource wiki for understanding how TinyMCE works and integrates.

Found a bug?

If you think you have found a bug, you can use the Tracker to report bugs to the developers.

And here is a simple table for you to play with.

Product	Cost	Really?
TinyMCE	Free	YES!
Plupload	Free	YES!

Path: h1 Words: 179

Druhá mocnina

Druhá mocnina čísla a je součin a.a.

Druhou mocninu vyjádříme jako $a^2 = a.a$.

a s horním indexem 2 čteme jako á na druhou.

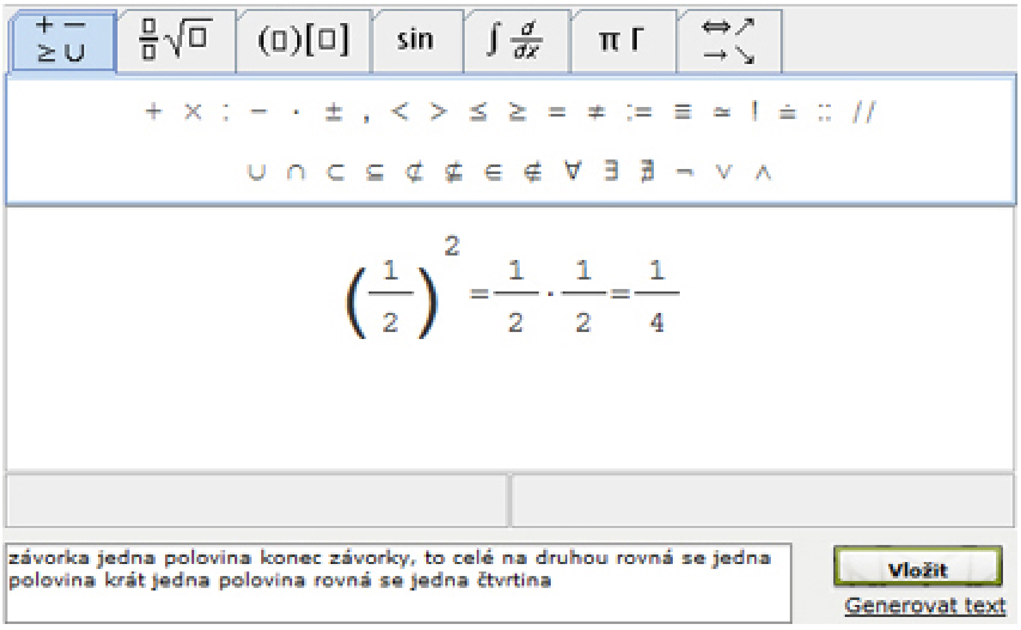
$6^2 = 6.6 = 36$

Figure 4.1: Left - TinyMCE editor [13], Right - TinyMCE editor in ARET administration

4.1 Managing HTML content

Editing a HTML content of a web page through an administrative interface is typically accomplished with the use of a Word-like WYSIWYG editor. This means that the editors provides functionality like the MS Word application, e.g. editing the textual information, their semantic structure and visual modification (changing color or font) can be done just by clicking on buttons. There are several popular choices of which editor to pick: *Asbru Web Content Editor* [11], *Toko Web Content Editor* [12] or *TinyMCE*. All of these editors provide basically the same functionality but only the last mentioned editor is open source and provides free and reliable HTML content management.

As was mentioned, TinyMCE (figure 4.1) is an opensource, platform independent, web based JavaScript HTML WYSIWYG editor. It has the ability to convert HTML TEXTAREA fields or other HTML elements to editor instances and is very easy to integrate into Content Management Systems. [13] It provides a very rich functionality for word processing and content management tools like *MCImageManager* or *MCFileManager* (these tools are not free and have to be payed for). The functionality is also very easily extendable through custom plugins. This particular feature is very important for the project ARET and will be further discussed in the following chapter.



$$\left(\frac{1}{2}\right)^2 = \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}$$

závorka jedna polovina konec závorky, to celé na druhou rovná se jedna polovina krát jedna polovina rovná se jedna čtvrtina

Figure 4.2: DragMath formula editor integrated into the project ARET administration

4.2 Managing mathematical formulas

One of the requirements for the project ARET is including a WYSIWYG formula editor into the administration, so that the administrators can insert and modify mathematical and physical formulas in the HTML code of the particular topic. The editor is required to be integrated into the web page (web administration), to be web-browser independent and to be able to work with different formula representations (mainly TEX and MathML for generating the images and textual representations of the given formula). There are several possible editors to choose, among the most popular and meeting the requirements are *DragMath*[14] and *MathDOX* [15]. Both editors are available under a open source licence and meet the given requirements, but due to performance issues in MathDOX, the DragMath editor (figure 4.2) was chosen. In addition, the DragMath editor provides the possibility of customization, i.e. the possibility to define new mathematical functions and operators.

Chapter 5

System architecture

This chapter and its following sections will describe thoroughly how the application for the project ARET is designed, which logical parts the application consists of and what program foundation was used to create the final application. Furthemore, the means of project management will be described, i.e. what program tools were used to manage the tasks and versioning.

5.1 Project management

Every software project of larger size and complexity requires some source control software and project management software. For building the application within the project ARET, *SVN (specifically TortoiseSVN 1.6)* and *Redmine 1.1* were used .

TortoiseSVN is an easy-to-use source control software for Microsoft Windows and possibly the best standalone Subversion client available. It is implemented as a Windows shell extension, which makes it integrate seamlessly into the Windows explorer. Since it's not an integration for a specific IDE it can be used with any development tool. [16]

Redmine is an opensource flexible project management web application, which provides creating wiki pages and creating tasks connected with SVN repository. It is cross-platform and cross-database. [17]

5.2 System architecture

The architecture of the final application can be viewed on figure 5.1. There are several logical parts, which have different functionality and cooperate with each other in order to read out the desired information contained in the website to the user (student). Another purpose is the administration of the content.

As can bee seen in the diagram 5.1, there are two separate types of users. The primary users are the students, for which the content is intended. The second type of users are the administrators, who take the responsibility for creating and maintaining the content.

The student can acces the information via any web browser (mostly *Internet Explorer*, *Mozilla Firefox*, *Chrome* or *Opera*), it is displayed on his computer monitor and simultaneously being read to him via the audio speakers. The part that is currently being read is also highlighted so the user sees the given part in the context of the webpage. This functionality is based on a *HTML preprocessor* (written in JavaScript and jQuery), which parses the HTML code of the given webpage, separates important parts of the page (parts, which are required to be read aloud), determines additional parameters according to the context (e.g. which type of voice to use) and then passes these information to the *audio player*.

This player (specifically *jPlayer* [18]) is a jQuery plugin for playing and controlling audio and video files on a webpage. It is inserted into the page and uses the information, received from the HTML preprocessor, to create a playlist of MP3 files. The script for creating the playlist is run just after the page is loaded, but does not create the complete playlist at once. For every textual segment of the page, there has to be sent a HTTP request to the *Web cache server* (the response to this request is also automatically being cached in the web browser), so sending all the requests at once would generate a heavy traffic load on the *TTS (Text To Speech)* server. The playlist is therefore created on the run - in the beginning, there are sent only a few requests, so that there is enough information to start navigating through the page (i.e. go to the next segment or title). Other requests are sent when one of the following events occurs: pressing the 'next button', 'previous button', 'next title button' or 'previous title button'. After a while of navigating, the complete playlist is therefore created.

The requests, that are being sent from the jPlayer to the server, contain textual information as well as some additional parameters (type of voice, tempo or format). When the server receives the request, it firstly checks the cached files in case that the same request has already been processed. According to the result, it either sends back a link, where the audio file is stored, or forwards the request to the *TTS generator* on the web cache server. The web cache server is primarily intended for generating audio files with TTS engine, but it has some additional engines at its disposal. It can be used also for generating images of formulas (TEXto image), generating texts (MathML to text) and some other, which will be described further in the implementation chapter.

The second type of users is the administrator. Users with administration privileges have access to the database and can add or edit the content via web administration interface. This administration contains tools for managing the database, where are stored authorized users, media files and topics. The topic can be accessed and edited via the TinyMCE editor, which is customized with own plugins. The most important plugin uses the DragMath formula editor for inserting mathematical and physical formulas into the HTML source code of the given topic. The DragMath editor also makes use of the before-mentioned web cache server, i.e. it uses the MathML-to-text and TEX-to-image processors to insert an image tag with the proper textual transcription into the HTML code.

When the administrator finishes creating or editing the current topic, he presses the *Save* button, so that the changes can be stored into the database. But there is one more step before applying the changes to the database. The HTML code is parsed and all the so called *inline formulas* are looked up and provided with a proper textual transcription. An inline-formula is a HTML *SPAN* tag with class attribute set to "inline-formula". This tag is supposed to contain a formula, which can be easily written in one line (e.g. $a+b=c$), so the administrator can avoid using DragMath editor and write formulas faster and more easily.

This phase of looking up inline formulas also makes use of the web cache server and its services (*formula-to-text* processor). After the code has been updated, it can be stored to the database and published to the students.

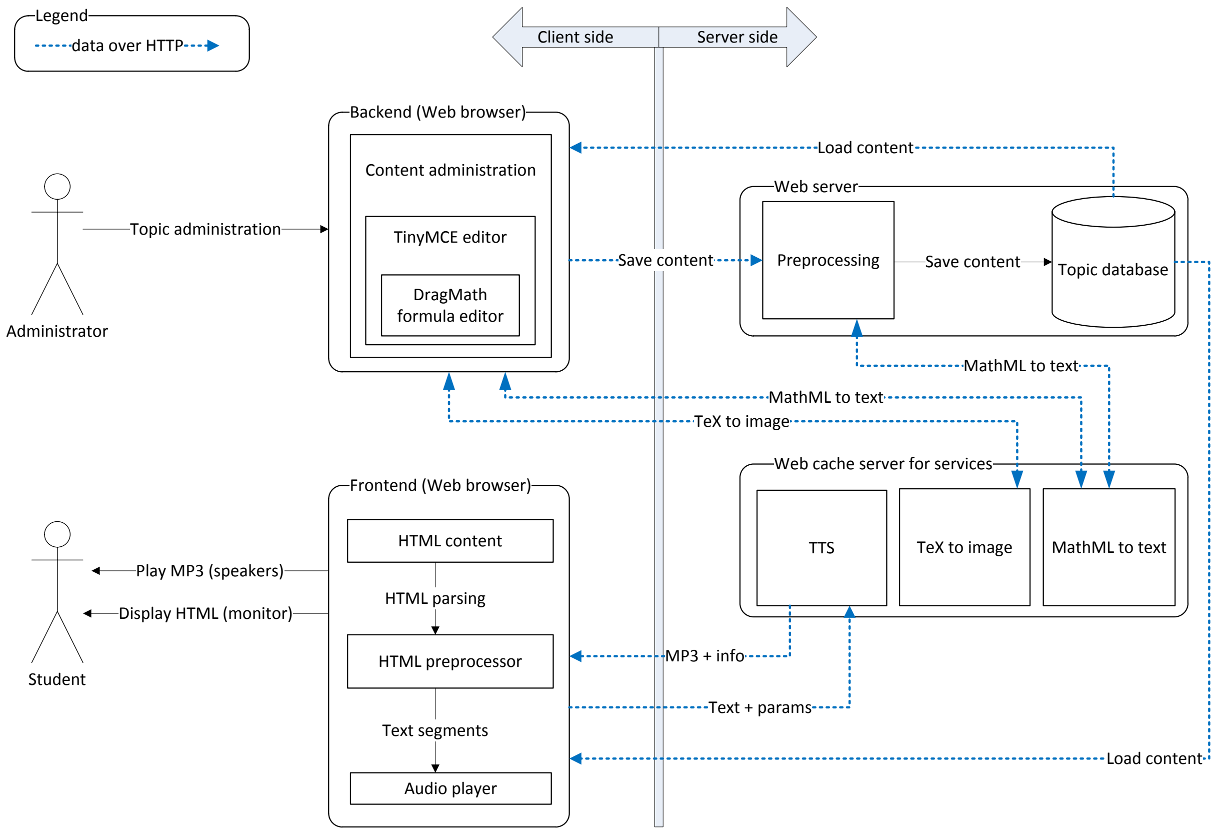


Figure 5.1: System architecture

Chapter 6

System implementation

This chapter and its sections will describe in detail the implementation of the final application. What parts were already available or in the process of development on the Department of Cybernetics. What and how was developed within this diploma thesis and how is everything incorporated together. There will also be thoroughly described some key aspects of functionality and how they were achieved.

6.1 Available CMS

The application for the project was not created completely from the scratch. The main reason for using a framework and the available plugins is the potential to start quickly developing the most problematic parts of the application and not focusing on the peripheral issues. As was mentioned earlier, the Department of Cybernetics on the University of West Bohemia has made available the usage of their CMS, which was already being developed for some time and for another project - dictionary of sign language (the interface can be seen in figure 6.1). More specifically, several plugins for the PHP framework Symfony were made available. They are intended for example for creating and maintaining a database of users, some for storing media files etc. The most important of them will be described in the following paragraphs as well as some plugins from the online Symfony community.

Symfony plugins and what they provide:

- sfDoctrineGuardPlugin - basic database schema and tools for managing users, user groups and user permissions.
- sfFormExtraPlugin - supplementary widgets and validators for basic Symfony forms.
- sfImageTransformPlugin - basic functionality and tools for managing images.
- sfTaskExtraPlugin - supplementary tasks for managing the Symfony project.

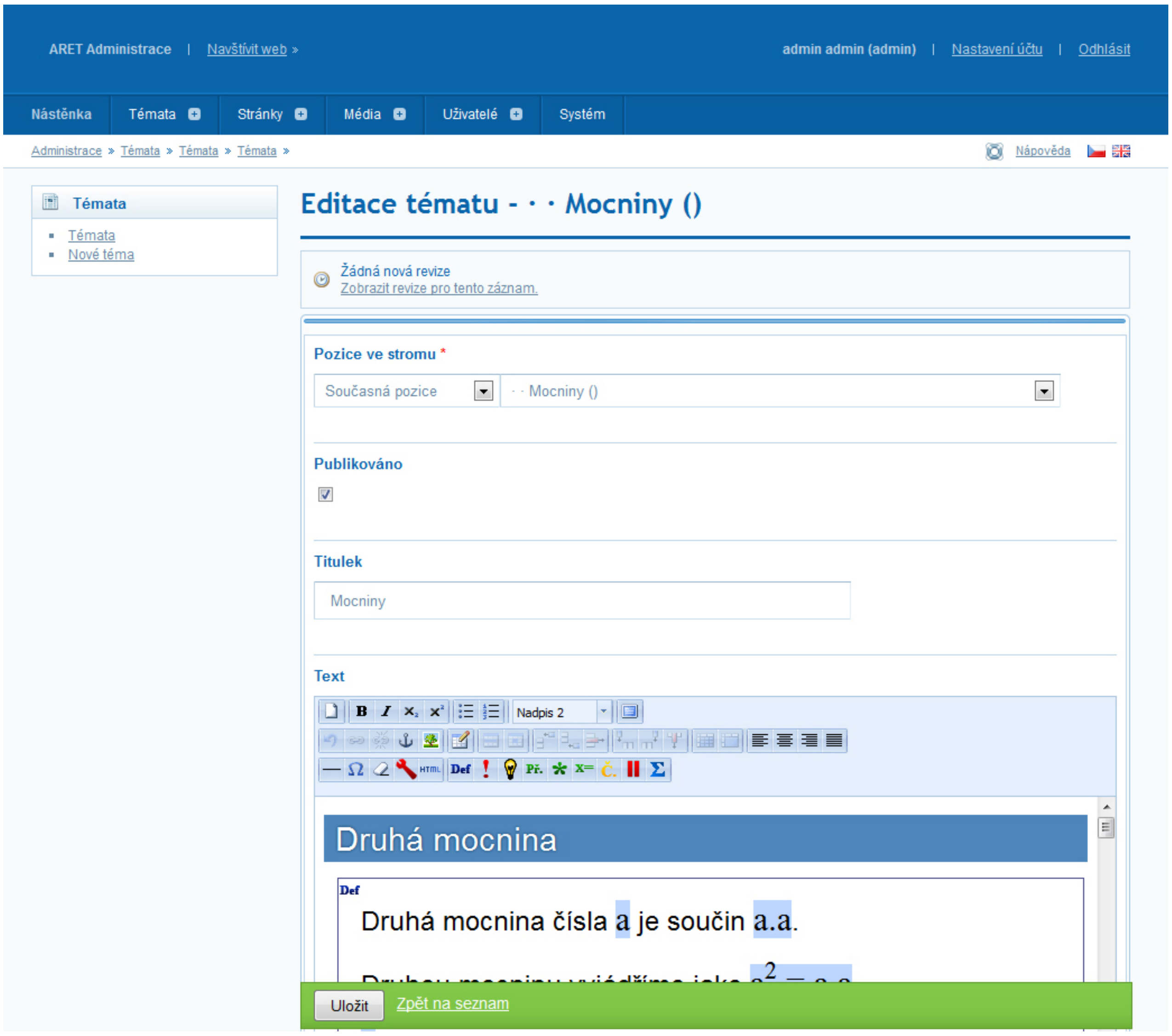
Plugins provided by the Department of Cybernetics:

- AdminPlugin - administrative interface for managing the database entries, its appearance (i.e. CSS files) and templates for automatically generated modules.
- AdminTaskPlugin - adds a possibility to append task for administrators to every database entry.
- CachePlugin - supplementary functionality of Symfony caching.
- CorePlugin - low level classes which extend the functionality of Symfony.
- DoctrinePlugin - low level classes which extend the functionality of Doctrine and a supplementary set of behaviors for database entries.
- DtCssPlugin - CSS preprocessor for extended formating of CSS files and their compilation.
- EmailPlugin - administration of email templates (for example editing the template of registration email or email with forgotten password).
- FormExtraPlugin - supplementary widgets and validators of forms.
- ImagePlugin - additional functionality for managing images.
- LockPlugin - provides functionality for locking the database entries in administration, so that any two different users can not edit the same entry at the same time.

- MediaLibrary2Plugin - provides an online file manager for administering files through administration. It is used mainly for managing media files (i.e. images).
- RevisionPlugin - adds the functionality for revision control (i.e. storing any changes made in any database entry and the possibility to allow only authorized users to accept and execute the changes).
- RoutingPlugin - extending the routing functionality (i.e. adding the possibilty to switch languages in URLs).
- SitemapPlugin - generating sitemaps.
- SlugPlugin - management of textual identifiers of database entries.
- SnippetPlugin - management and presentation of short text blocks placed anywhere in the web.
- TinyMcePlugin - adds the basic functionality of TinyMCE editor into the administration interface.
- UserPlugin - further enhancement of the sfDoctrineGuardPlugin for managing users, user groups and user permissions.
- WebResourcesPlugin - static resources for web (i.e. CSS files, JavaScripts and images).
- WorkflowPlugin - adds the possibility to define different conditions of database entries and transitions between them (e.g. converting a topic from a *hidden* state to a *published* state).

6.2 Project specific plugins

The previous section described all the plugins used for creating the main parts of the application, but for the required functionality, defined in the project ARET, there had to


ARET Administrace | Navštívit web »

admin admin (admin) | Nastavení účtu | Odhlásit

Nástěnka Témata Stránky Média Uživatelé Systém

Administrace » Témata » Témata » Témata »

Nápověda

Témata
- Témata
- Nové téma

Editace tématu - · · Mocniny ()

Žádná nová revize
Zobrazit revize pro tento záznam.

Pozice ve stromu *
Současná pozice · · Mocniny ()

Publikováno
✓

Titulek
Mocniny

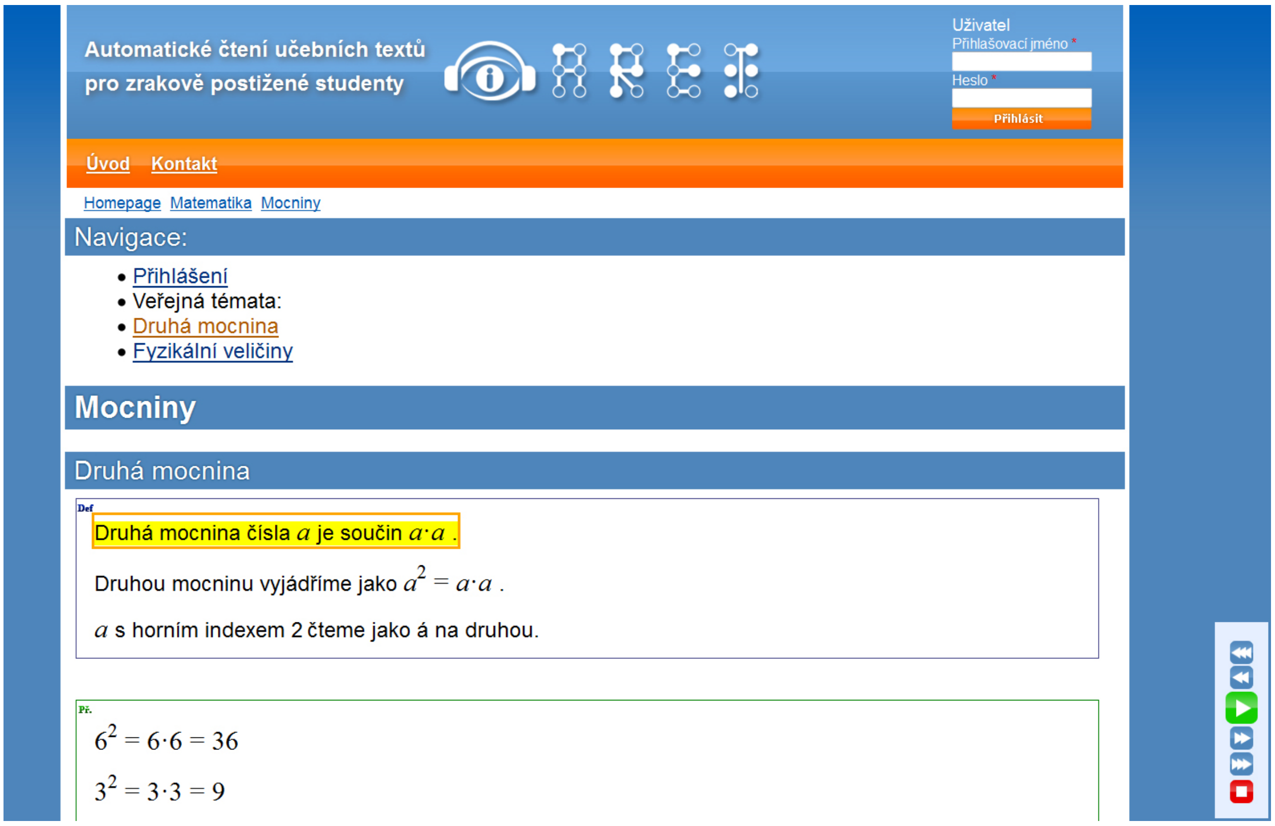
Text
Nadpis 2

Druhá mocnina

Def
Druhá mocnina čísla a je součin a.a.

Uložit Zpět na seznam

Figure 6.1: Backend - the administration interface.



Automatické čtení učebních textů pro zrakově postižené studenty

Navigace:

- Přihlášení
- Veřejná témata:
- Druhá mocnina
- Fyzikální veličiny

Mocniny

Druhá mocnina

Def

Druhá mocnina čísla a je součin $a \cdot a$.

Druhou mocninu vyjádříme jako $a^2 = a \cdot a$.

a s horním indexem 2 čteme jako á na druhou.

Př.

$6^2 = 6 \cdot 6 = 36$

$3^2 = 3 \cdot 3 = 9$

Figure 6.2: Frontend - public interface for reading topics.

be created some other custom plugins. Creating these plugins was the main goal of this diploma thesis as well as configuring all the provided plugins and maintaining the whole project. The public interface (frontend) was also created and configured within these plugins (figure 6.2). The four following plugins were created within this diploma thesis:

6.2.1 Project configuration - mcAretPlugin

This particular plugin serves as a suplement to the general configuration of the whole project. There is mainly placed the configuration of the structure of the administration (i.e. the structure of the navigation menus). In addition to that, there are located the main CSS files, which contain the definition of the visual aspect of the frontend. No database features are present here.

6.2.2 Creating mathematical formulas - mcTinymceDragmathPlugin

This plugin enhances the functionality of the TinyMCE editor via custom plugins for this editor. These plugins are following (ordered by appearance on the figure 6.3):

- mcCorrectHtml - This plugin makes use of another service which is available on the *web cache server*. After clicking the appropriate button (the red wrench) a request is send to the server with the whole HTML code. This code is then transformed with the use of the *correct_html* service. This particular script tries to repair any possible errors in the HTML code and sends it back to the TinyMCE editor.
- mcDefinition - This plugin inserts a *DIV* tag with *class* attribute set to "definition". This tag is displayed in frontend with a blue border and when the user gets to this segment a special sentence is added to the players playlist: "Zapamatujte si.". The whole content of this tag is also read by a different voice.
- mcWarning - This plugin inserts a *DIV* tag with *class* attribute set to "warning". This tag is displayed in frontend with a red border and when the user gets to this segment a special sentence is added to the players playlist: "Pozor!". The whole content of this tag is also read by a different voice.
- mcInfo - This plugin inserts a *DIV* tag with *class* attribute set to "mcinfo". This tag is displayed in frontend with a green border on the left and when the user gets to this segment a special sentence is added to the players playlist: "Pomůcka.". The whole content of this tag is also read by a different voice.
- mcExample - This plugin inserts a *DIV* tag with *class* attribute set to "example". This tag is displayed in frontend with a green border and when the user gets to this segment a special sentence is added to the players playlist: "Příklad.". The whole content of this tag is also read by a different voice.
- mcSolution - This plugin inserts a *DIV* tag with *class* attribute set to "solution". This tag is displayed in frontend with a green border and when the user gets to this segment

a special sentence is added to the players playlist: "Řešení.". The whole content of this tag is also read by a different voice. When viewed in frontend, all the content in this tag is hidden and replaced by a button. Only after pressing this button, the content of the solution is revealed. This forces the student to think about the example without being immediately presented with the solution.

- mcFormula - This plugin serves for marking the inline formulas, which will be further processed during saving the content into the database. These formulas are to be updated with a textual transcription (in the *alt* attribute) and a formated form (in the *name* attribute). An example of such formating can be viewed in figure 6.5.

- mcReadAs - The purpose of this plugin is adding a correct phonetic transcription for non standard words. For example the name "Newton" is in Czech language required to be read like "Njůtn". The required transcription can be therefore added just after the word and marked with the use of this plugin. The word directly before the marked one is then skipped and replaced with its transcription, when the HTML processing in frontend is in progress.

- mcPause - This plugin can be used for marking the place where the reading of the document is required to stop and wait for the user to fully understand the content. The user is then supposed to start the reading by pressing the *Play button*.

- dragmath - This is the most important plugin for the project ARET. This plugins provides the functionality for inserting even very complex mathematical or physical formulas. This plugin makes use of the *DragMath* formula editor, which is available as an opensource Java applet. This editor has been upgraded with custom functions and operators. It has also been adapted so that it can be used in collaboration with the *TinyMCE editor*. When the appropriate button is pressed in the TinyMCE editor, a popup window appears (see figure 4.2). This window contains the DragMath editor, a button for saving the formula and a *textarea* which contains the preview of the textual transcription. This transcription can be generated by clicking on the link that is labeled "Generovat text". By pressing the "Vložit" button, the formula is inserted into the HTML code of the topic along with other additional formats.

All of these information are located in one *IMG* tag. The textual transcription is stored in the *alt* attribute. The *src* attribute contains all the information required for generating the image (which TEXtemplate to use, the desired image resolution and the TEXnotation of the formula). An example of such an URL, pointing to an image of formula *a+b*:

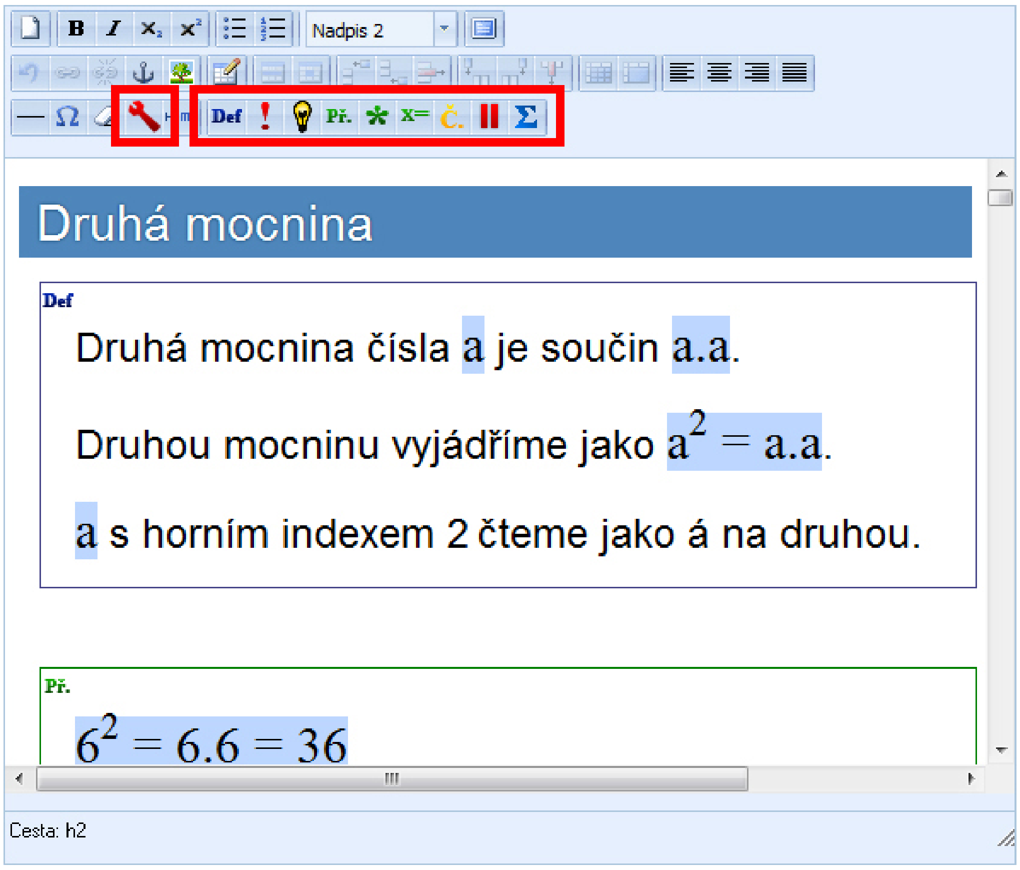
```
http://tts.zcu.cz/mathTex.png?generator[template]=aretmath
&generator[dpi]=200&generator[tex]=a+b
```

Other information is stored in following *HTML5 data attributes*:

- The attribute *data-base64* contains a serialized (into a base64 encoded string) Java object which represents the given formula and can be used to load the formula back into the DragMath editor. The editor is able to load the formula in other formats but loading for example a MathML or TEXrepresentation proved to be inaccurate and in some cases was loaded different formula than was originally generated. Using the serialized object is the only reliable way of storing the formula and loading it correctly.
- The attribute *data-mathml* contains the MathML representation of the formula. This can be further used for regenerating the textual transcriptions of all formulas at once by running a script on the server.
- The attribute *data-tex* contains a backup of the TEXtranscription. This format is used for generating the image of the formula.

6.2.3 Reading web pages - mcWebTtsPlugin

This plugin is responsible for parsing the HTML content, selecting the important parts and reading them to the user. The parsing is accomplished with the use of the *jQuery* library.



Druhá mocnina

Def

Druhá mocnina čísla a je součin a.a.

Druhou mocninu vyjádříme jako $a^2 = a.a$.

a s horním indexem 2 čteme jako á na druhou.

Př.

$6^2 = 6.6 = 36$

Figure 6.3: TinyMCE plugins for project ARET (in red frames).

The HTML code of each topic, being generated by the TinyMCE editor and additionaly chcecked by the *correct_html* script, can be considered to be valid and well formatted. So seeking the important parts of the content which are to be read out is not very problematic, especially with the use of JavaScript and the jQuery library.

With the use of the jQuery selectors, the HTML tags which contain the important texts are selected (*A*, *H1*, *H2*, *H3*, *H4*, *LI* and *P*). Each one of these tags is looked out and the text, which is contained in it, is surrounded by a *SPAN* tag with *class* attribute set to ”ttsText”. The *P* and *LI* tags are further checked if they contain an image or a link, so they can also be marked. Marking those segments is important for synchronization of reading and highlighting during reading.

When the marking of the textual segments is completed, additional processing can be started. This means once again going through all of the tags in the document, determining their importance for reading and, according to that, setting additional parameters and start creating the playlist for the jPlayer. The important tags are:

- *DIV* - these tags can represent the project specific templates (i.e. *info*, *definition*, *example*, *solution* and *warning*) and thus additional sentences need to be added to the playlist. The content of those templates is also required to be read by a different synthetic voice, so additional parameters are set.

- *OL*, *UL* - when this tag is found, the list-item-counter is reseted back to 0 so the list items (LI tags) can be counted from the begining.

- *LI* - when this tag is found, the list-item-counter is incremented and the appropriate sentence is added to its content (e.g. the words ”Za druhé” are added to the text of the second list item).

- *IMG* - when this tag is found, it is replaced by the content of its *alt* attribute so the so the surrounding HTML code contains only the textual transcription and not the tag itself.

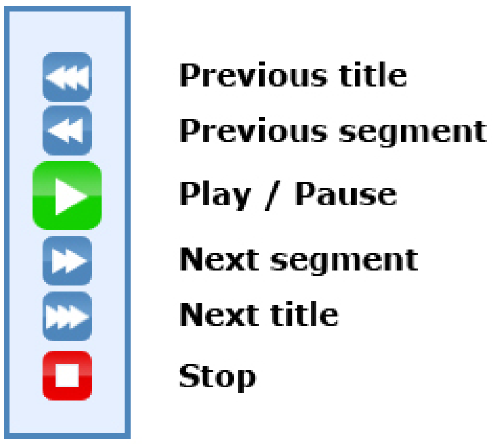


Figure 6.4: The audio player and its functions.

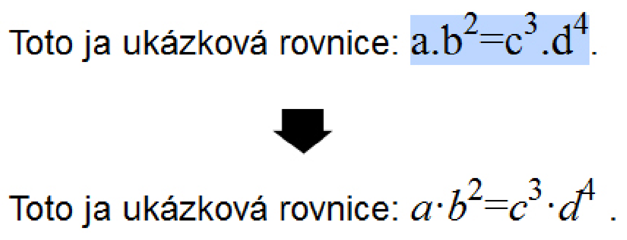
- *SPAN* - these tags are considered only if their *class* attribute is set to "ttsText". They contain the previously marked information which are to be sent to the TTS server. This requires further modification of the marked tags, i.e. getting rid of all the residual HTML tags, processing the *read-as* tags and replacing any possible tags, with a not empty *alt* attribute, with the attribute content. This ensures the fact that whatever text is being sent to the TTS server does not include any unwanted characters and consists of only pure text.

All of these tags are also updated with attribute *tabindex*, which enables the tags to be highlighted and have the focus of the page being set on them.

After the parsing is finished and the playlist (now containing the text and additional parameters) is created, the jPlayer starts sending those information to the TTS server as was described in chapter 5.

6.2.4 Topic database - mcCourseBookPlugin

This plugin contains everything that is somehow connected to storing and managing the topics. Mainly, there is the configuration of the database for storing the HTML code of


Toto ja ukázková rovnice: $a.b^2=c^3.d^4$.

Toto ja ukázková rovnice: $a \cdot b^2=c^3 \cdot d^4$.

Figure 6.5: Transformation of inline formula from backend to frontend.

each topic - definition of the columns, thir relations to other database tables and behaviors which further define the topics functionality.

There is implemented the functionality described in the previous chapter concerning the processing of the HTML code before saving it to the database (i.e. updating the inline formulas with textual transcription and formated form for better visual understanding). An example of this can be viewed in figure 6.5. At the top of the figure is an inline formula as can be seen in the TinyMCE editor and highlighted for further processing. The bottom line is how the formula looks like in the frontend. The textual information is hidden, but the formating has changed (the period has been replaced with the dot in the vertical center of the line).

This plugin also contains a JavaScript which is responsible for replacing the content of any solution template by a button so that the solution of the given example is hidden and can be viewed only after pressing this button.

6.3 Web services

As was depicted in the chapter 5 and on figure 5.1, the developed application makes use of various web services (Web API). Some of these services were in the early stages of the project developed independently and were placed in the same folder as the application. But with further development, all of the services required for the project were united on one place and under the same domain (*tts.zcu.cz*). All of the scripts can be therefore called on one place and with similary strucured parameters. Every result of any used script is also being cached so the system is not asked to run the script every time. A security layer was also developed.

The services used by the projects' application are:

- LightTts - This web service uses the lightweight TTS server developed on the Department of Cybernetics. It provides the text-to-speech synthesis as a web service. Possible parameters are: *text*, *tempo*, *speed*, *pitch*, *volume*, *format* and *voice*.
- MathML2Text - This service provides the transcription from MathML to text via a custom Python cript provided by the Department of Cybernetics.
- Formula2Text - This service provides the transcription of formulas (*inline formulas* written in HTML) to text via a custom Python cript provided by the Department of Cybernetics.
- FormatFormula - This service provides the transcription of inline formulas to a better formated notation via a custom Python cript provided by the Department of Cybernetics.
- HtmlCorrector - This service provides the possibility of checking the HTML of a whole topic for possible errors and mistakes via a custom Python cript provided by the Department of Cybernetics.

Chapter 7

Evaluation and conclusion

The application, which was developed within the project ARET, is by now in the second year of its development (including the primary analysis and consultations with the teachers of the partner school). Although the application is still being developed, it is already being used withing the math and physics classes in the mentioned primary school. This usage also provides very useful information for further upgrades of the application and thus making it even better and more easy to use by the target users, i.e. the visualy impaired students.

The application is also being revised by a visualy impaired student of the University of West Bohemia, who provides further counseling and recommendations. For example, which keyboard shortcuts are suitable for the use on the web page, that are not in conflict with any possible screen reader, which the students can use. Suggestions like this are very important for the development and provide the possibility for the application to be useful and actually fulfill the aim of the whole project, which is helping the students in their studies.

The resulting functionality, which is now available, is a result of several stages of development. Because of the variable needs of the teachers and students, the application has undergone many changes and even now, some changes are planned. Since the application is not yet fully finnished, this diploma thesis can not incorporate all of the final functionality, but for now, the application is working as was firstly desired and thus accomplishes its objective.

The application in its current state has also been presented on several conferences (ULD 2011 and INSPO 2011) and thus was mentioned in corresponding papers ([19], [20]). Two more papers were also submitted (conferences TSD and ASSETS) and have not been accepted yet.

Any future upgrades can also possibly be incorporated in another project of the Department of Cybernetics, which is now in the state of approval by the European union - ARET2.

List of abbreviations

- ARET - Automatic Reading of Educational Texts (the official name of the project)
- HTML - HyperText Markup Language
- CMS - Content Management System
- MathML - Mathematical Markup Language
- WYSIWYG - What You See Is What You Get
- URL - Uniform Resource Locator
- XHTML - Extensible HTML
- XML - Extensible Markup Language
- AJAX - Asynchronous JavaScript and XML
- CSS - Cascading Style Sheets
- JVM - Java Virtual Machine
- HTTP - Hypertext Transfer Protocol
- JSP - Java Server Pages
- ASP - Active Server Pages
- CLR - Common Language Runtime

- DRY - Don’t Repeat Yourself
- CRUD - Create, Read, Update, Delete
- PHP - PHP: Hypertext Preprocessor (recursive abbreviation)
- MVC - Model View Controller
- ORM - Object Relational Mapper
- DBAL - Database Abstraction Layer
- OO - Object Oriented
- SQL - Structured Query Language
- DQL - Doctrine Query Language
- HQL - Hybernate Query Language
- SVN - Subversion
- IDE - Integrated development environment
- TTS - Text To Speech
- SEO - Search Engine Optimization

Bibliography

[1] Screen magnifiers [online]. http://www.pristupnost.cz/zvetsovaci-softwarove-lupy/, 2011.

[2] Wikipedia, the free encyclopedia [online]. http://www.wikipedia.org, 2011.

[3] Lambda - linear access to mathematics for braille device and audio-synthesis [online]. http://www.teiresias.muni.cz/czbraille8/, 2011.

[4] Spoken web - reading of web pages [online]. http://www.spoken-web.com, 2011.

[5] Browsealoud - reading of web pages [online]. http://www.browsealoud.com, 2011.

[6] Mathplayer - reading of math formulas [online]. http://www.dessci.com/en/products/mathplayer/, 2011.

[7] Python documentation [online]. http://docs.python.org, 2011.

[8] Ruby [online]. http://www.ruby-lang.org, 2011.

[9] Symfony [online]. http://www.symfony-project.org, 2011.

[10] Doctrine orm [online]. http://www.doctrine-project.org/, 2011.

[11] Asbru web content editor [online]. http://www.asbrusoft.com/, 2011.

[12] Toko web content editor [online]. http://toko-contenteditor.pageil.net/, 2011.

[13] Tinymce [online]. http://tinymce.moxiecode.com/, 2011.

[14] Dragmath formula editor [online]. `http://www.dragmath.bham.ac.uk/`, 2011.

[15] Mathdox - math editor [online]. `http://www.mathdox.org`, 2011.

[16] Tortoise svn [online]. `http://tortoisesvn.net/`, 2011.

[17] Redmine [online]. `http://www.redmine.org/`, 2011. xxx.

[18] jplayer - jquery audio player [online]. `www.jplayer.org/`, 2011.

[19] Jindřich Matoušek, Michal Campr, Zdeněk Hanzlíček, and Martin Grůber. Automatic reading of educational texts for vision impaired students. In *ULD - Universal Learning Design, Brno*, 2011.

[20] Jindřich Matoušek, Michal Campr, Zdeněk Hanzlíček, and Zdeněk Krňoul. Aret – automatické čtení učebních textů pro zrakově postižené studenty. In *INSPO - Internet a informační systémy pro osoby se specifickými potřebami, Praha*, 2011.