Final Career Project

Implementing the FODITIC functions as a standard module of the Claroline open source E-learning platform

To get the title “Ingeniería Técnica en Informática de Gestión” from UPC (Universidad Politécnica de Catalunya).

Authors: David Bayo Nieto
Director: Marc Lobelle
First Reader: Frederic Minne
Second Reader: Philippe Dekimpe
ABSTRACT

In software engineering, a web application is an application delivered to users from a web server over a network such as the World Wide Web or an intranet. Web applications are popular due to the ubiquity of the web browser as a client, sometimes called a thin client.

The Claroline platform is an eLearning and eWorking web application, allowing teachers to build effective online courses and to manage learning and collaborative activities on the web.

Also, Claroline is Open Source software; therefore the platform can be modified, so we can extend or improve its functionality. A module is an installable package that adds either new restrictions or functionalities to the Claroline platform.

Foditic is a website that uses an old release of the Claroline platform, modified to add new functionalities, improving the nature of the learning material itself.

Two new Claroline modules have been created. They contain the new Foditic functions implemented to extend the functionality of the platform, adapted for unmodified Claroline servers. Thus, current and future releases of Claroline can be installed anywhere with the new features and enhancements that Foditic has contributed to its own website.
ACKNOWLEDGEMENTS

This has been a year of great academic and personal growth.

On the academic side, I want to thank my supervisor Mr. Marc Lobelle for his continuous support, always finding time to sit down with me whenever I needed it. I also want to express my gratitude to the rest of the University's members and staff that I have met throughout this academic year.

On the personal side, I want to acknowledge the importance to me of my parents, Raul Bayo and Josefa Nieto, my girlfriend, Mireia Morales and all my friends. This has been my first year away from home, and possibly many more will come. No matter where I may be, I will always know where they are.

I will take many things with me from Louvain-la-Neuve, but the most important ones are the important acquired knowledge and the strong friendships achieved throughout this year.

¡Gracias a todos por hacer de éste un año para recordar!
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td></td>
<td>I</td>
</tr>
<tr>
<td>ACKNOWLEDGEMENTS</td>
<td></td>
<td>II</td>
</tr>
<tr>
<td>TABLE OF CONTENTS</td>
<td></td>
<td>III</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>0.</td>
<td>INTRODUCTION AND STRUCTURE</td>
<td>1</td>
</tr>
<tr>
<td>1.</td>
<td>FODITIC</td>
<td>2</td>
</tr>
<tr>
<td>1.1.</td>
<td>THE FODITIC FUNCTIONS INTENDED FOR STUDENTS</td>
<td>3</td>
</tr>
<tr>
<td>1.2.</td>
<td>THE FODITIC FUNCTIONS INTENDED FOR COURSE ADMINISTRATORS</td>
<td>5</td>
</tr>
<tr>
<td>1.2.1.</td>
<td>Fodi_import</td>
<td>7</td>
</tr>
<tr>
<td>1.2.2.</td>
<td>Fodi_export</td>
<td>9</td>
</tr>
<tr>
<td>1.2.3.</td>
<td>Fodi_gloss</td>
<td>9</td>
</tr>
<tr>
<td>1.3.</td>
<td>IMPLEMENTATION</td>
<td>11</td>
</tr>
<tr>
<td>1.3.1.</td>
<td>INTRODUCTION</td>
<td>11</td>
</tr>
<tr>
<td>1.3.2.</td>
<td>Fodi claro_modifs</td>
<td>11</td>
</tr>
<tr>
<td>1.3.2.1.</td>
<td>Claro_init_footer</td>
<td>11</td>
</tr>
<tr>
<td>1.3.2.2.</td>
<td>Claro_init_banner</td>
<td>12</td>
</tr>
<tr>
<td>1.3.2.3.</td>
<td>Document</td>
<td>13</td>
</tr>
<tr>
<td>1.3.2.4.</td>
<td>Courses</td>
<td>14</td>
</tr>
<tr>
<td>1.3.3.</td>
<td>Fodi_lang</td>
<td>14</td>
</tr>
<tr>
<td>1.3.4.</td>
<td>Fodi_scripts</td>
<td>15</td>
</tr>
<tr>
<td>1.3.4.1.</td>
<td>imgs</td>
<td>15</td>
</tr>
<tr>
<td>1.3.4.2.</td>
<td>Fodi_import</td>
<td>15</td>
</tr>
<tr>
<td>1.3.4.3.</td>
<td>Fodi_import_lib</td>
<td>17</td>
</tr>
<tr>
<td>1.3.4.4.</td>
<td>Fodi_export</td>
<td>17</td>
</tr>
<tr>
<td>1.3.4.5.</td>
<td>Fodi_bilingual.inc.php</td>
<td>17</td>
</tr>
<tr>
<td>1.3.4.6.</td>
<td>Fodi_cartouche</td>
<td>18</td>
</tr>
<tr>
<td>1.3.4.7.</td>
<td>Fodi_cartouche_BL</td>
<td>18</td>
</tr>
<tr>
<td>1.3.4.8.</td>
<td>Fodi_comment_aide.txt</td>
<td>18</td>
</tr>
<tr>
<td>1.3.4.9.</td>
<td>Fodi_comment_config.php</td>
<td>18</td>
</tr>
<tr>
<td>1.3.4.10.</td>
<td>Fodi_comment.inc.php</td>
<td>18</td>
</tr>
<tr>
<td>1.3.4.11.</td>
<td>Fodi_comment_only.inc.php</td>
<td>18</td>
</tr>
<tr>
<td>1.3.4.12.</td>
<td>Fodi_comment.js</td>
<td>18</td>
</tr>
<tr>
<td>1.3.4.13.</td>
<td>Fodi_comment.php</td>
<td>19</td>
</tr>
<tr>
<td>1.3.4.14.</td>
<td>Fodi_footer.inc.php</td>
<td>19</td>
</tr>
<tr>
<td>1.3.4.15.</td>
<td>Fodi_gloss_auto.php</td>
<td>19</td>
</tr>
<tr>
<td>1.3.4.16.</td>
<td>Fodi_gloss</td>
<td>19</td>
</tr>
<tr>
<td>1.3.4.17.</td>
<td>Fodi_gloss.inc.php</td>
<td>20</td>
</tr>
<tr>
<td>1.3.4.18.</td>
<td>Fodi_qr.inc.php</td>
<td>20</td>
</tr>
<tr>
<td>1.3.4.19.</td>
<td>Foditic.js</td>
<td>20</td>
</tr>
<tr>
<td>1.3.4.20.</td>
<td>Modeemploi_foditic.doc</td>
<td>20</td>
</tr>
<tr>
<td>2.</td>
<td>CLARONOE</td>
<td>21</td>
</tr>
<tr>
<td>2.1.</td>
<td>ABOUT CLARONOE</td>
<td>21</td>
</tr>
<tr>
<td>2.2.</td>
<td>EASY HANDLING</td>
<td>22</td>
</tr>
<tr>
<td>2.3.</td>
<td>MODULES</td>
<td>22</td>
</tr>
<tr>
<td>3.</td>
<td>THE “FACTORY” MODULE</td>
<td>23</td>
</tr>
<tr>
<td>3.1.</td>
<td>INSTALLATION OF THE MODULE FACTORY</td>
<td>23</td>
</tr>
<tr>
<td>3.2.</td>
<td>ACTIVATION OF THE MODULE FACTORY</td>
<td>25</td>
</tr>
<tr>
<td>3.3.</td>
<td>CHANGE THE RIGHT OF THE MODULE FACTORY</td>
<td>26</td>
</tr>
<tr>
<td>3.4.</td>
<td>AFTER THE INSTALLATION</td>
<td>28</td>
</tr>
<tr>
<td>3.5.</td>
<td>MAIN PAGE</td>
<td>29</td>
</tr>
<tr>
<td>3.6.</td>
<td>FUNCTIONS</td>
<td>30</td>
</tr>
<tr>
<td>3.6.1.</td>
<td>Claro_import</td>
<td>30</td>
</tr>
<tr>
<td>3.6.1.1.</td>
<td>Common tags</td>
<td>30</td>
</tr>
<tr>
<td>3.6.1.2.</td>
<td>Bilingual tags</td>
<td>31</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

Fig 1.1. Cartridge of a bilingual document. [2] .................................................. 3
Fig 1.2. Breadcrumb of the document “chap4.1” [2] ............................................... 4
Fig 1.3. Breadcrumb of the document “chap4.2” [2] ............................................... 4
Fig 1.4. Main page of “Documents et Liens” of the Foditic website [1] ....................... 8
Fig 1.5. Functions of the page “Documents et Liens” of the Foditic website [1] ........... 9
Fig 1.6. Main page of the “Créer un Glossaire” of the Foditic website [1] ................. 10
Fig 1.7. Footer of the Foditic website [2] ............................................................. 11
Fig 1.8. Header of the Foditic website [2] ............................................................ 12
Fig 1.9. Cartridge of a bilingual document 2. [2] ................................................... 18
Fig 1.10. Window to write the comments. [2] ....................................................... 19

Fig 3.1. Platform administration main page of Claroline ........................................ 23
Fig 3.2. List of tools in the main page of Modules before the installation of the “Factory” module ............................................................... 24
Fig 3.3. Options selected before the installation of the “Factory” module ................. 24
Fig 3.4 List of tools in the main page of Modules after the installation of the “Factory” module ............................................................... 25
Fig 3.5. Main page of a course ............................................................................ 25
Fig 3.6. Main page of the “Edit tool list” of a course ............................................. 26
Fig 3.7. Main page of a course ............................................................................ 27
Fig 3.8. Alert when a student tries to access to “Factory” module ......................... 29
Fig 3.9. Main page of “Factory” module after to use “claro_import” ...................... 33
Fig 3.10. Main page of “Factory” module after to use “claro_export” ...................... 34
Fig 3.11. Glossary example ................................................................................. 34
Fig 3.12. Main page of the “Glossary” function .................................................... 35
Fig 3.13. Main page of the “Add entry to course tool list” function ....................... 36
Fig 3.14. Main page of the “Clone this course” function ....................................... 38
Fig 3.15. Step 2 of the “Clone this course” function ............................................. 39
Fig 3.16. Step 3 of the “Clone this course” function ............................................. 40
Fig 3.17. Step 4 of the “Clone this course” function ............................................. 41

Fig 4.1. Platform administration main page of Claroline ........................................ 43
Fig 4.2. List of tools in the main page of Modules before the installation of the “Clone a course” module ......................................................... 43
Fig 4.3. List of administration tools after to install the “Clone a course” module ........ 44
Fig 4.4. Main page of the “Clone a course” module ............................................. 44
Fig 4.5. Step 2 of the “Clone a course” module ................................................... 45
Fig 4.6. Step 3 of the “Clone a course” module ................................................... 46
Fig 4.7. Step 4 of the “Clone a course” module ................................................... 47
Fig 4.8. Step 5 of the “Clone a course” module ................................................... 48
0. Introduction and structure

This project has been done by the student David Bayo Nieto with the help of the coordinator Marc Lobelle to get the title of Bachelor of Computing Sciences.

The goal of this Bachelor’s project is to implement the main functions added to the Claroline platform of the (2005 version) Foditic server as standard Claroline (2010 version) modules.

The software has been developed in a web environment, to be available to anyone on the Internet. Consequently, it can also serve as an example of developing a generic web application, using the php technology and MySQL database.

The text consists of X chapters in which I explain the necessary steps for any user to install and use these modules, adding the necessary details to work with them.

The first chapter, I explain the necessary information about Foditic, explaining the goal of this website and the specific functions it provides beyond the classical Claroline functions. Besides, I explain the details of such functions, explaining first the functions intended for the students and then, the functions intended for the course administrator. Additionally, I explain how each function is implemented and the functionality of the php files used.

In the second chapter, I explain the use of Claroline API functions necessary to adapt the functions as new modules of the Claroline platform. Also, I explain the steps to create a server with such platform, explaining also the requirements to do it and how to use modules on the Claroline platform. Besides, I explain in detail the modules of Claroline and how install and active them.

In the third chapter, I explain a new module called Factory, explaining mainly how it is adapted on the Claroline platform. Furthermore, I explain in detail the functions of this module and the steps to use these functions.

In the fourth chapter, I follow the same structure than the third chapter to explain the other new module “Clone a course”.

Finally, in the last chapter, I write a conclusion about the project, explaining the advantages and disadvantages of the Claroline platform to adapt the Foditic functions and explaining the personal opinion about this project.


1. Foditic

The Foditic website is a support for distance education in some fields of computing science and the language education. It is based on the Claroline distance learning environment.

Several services are added on the Foditic servers that are not on a standard Claroline server.

Basically, the difference between standard Claroline and Foditic is in the nature of the learning material itself. In Claroline, learning documents can be in any passive document format, usually html or pdf.

Besides these documents, Claroline includes a set of learning tools that use php programs and a MySQL database.

In Foditic, documents are php programs using also the MySQL database for some functions. They are thus of the same nature as the Claroline tools and the Claroline platform itself.

Since they are not simple documents but active entities, they can include various services related to the presentation of the documents and allowing each student to annotate the document for himself.

Using programs has the advantage of enabling the inclusion of new services but also could put the platform at risk; a course including errors in such active documents could crash the platform, which is unacceptable. Therefore Claroline refuses uploading php programs as documents.

In order to avoid this risk in Foditic, the php documents are not written but automatically generated starting from html document including special Foditic tags. If the generator is error free, so will also be the generated document. Another advantage of generating automatically the php document is that no programming skills are required to write the “source” html documents.

This document will first present the new functions available in Foditic php documents as they are provided to the learners and then the tools provided to professors to generate the active documents.

The current implementation of Foditic existing before this project and unwanted characteristics:

- When it was originally implemented, Claroline was not really modular so several Claroline programs hard to be modified to accommodate the Foditic functions.
• The tools for generating the programs are hybrid. Some functions are available in a modified “document and links” Claroline tool. But the most up to date version of the transformation program is a stand-alone php program that has to be run manually on the server; it requires logging on the server (with ssh) which involves risks for the latter.

The purpose of this project is to modify the Foditic software to make it a Claroline module compliant with Claroline module rules, using the Claroline API and not dependent on Claroline implementation. All functions should be accessible through the “professor” code interface to Claroline and may not put the platform at risk.

1.1. The Foditic functions intended for students

The Foditic platform is designed to increase the learning comfort of the students. Studying on-line should be as comfortable as studying in a book and provide as well learning helps possible on a computer but not in a head. When the Foditic system displays a php page including Foditic functions, it shows automatically an introductory table or cartridge at the top of each php page, like the next picture.

![Cartridge of a bilingual document](image)

The student can:

• Annotate for himself the document in the spaces provided by the teacher in the document itself. Add personal notes to this text by clicking on the little notebooks ; the notebooks that include your comments get a yellow color:
• Choose to visualize the comments of the professor and their personal notes in pop-up windows or in-line. For switching between these modes, the student has to select the first check-box and reload.
• Choose to visualize the pictures in pop-up windows or in-line. For switching between these modes, he has to select the second check-box and reload.
• See the comments written by the teacher by clicking on: 🆕 (if the student chooses pop-up windows).
• See the pictures attached by the teacher by clicking on: 🌟 (if the student chooses pop-up windows).
• Get a printable version of the chapter including this document in a new temporary window by clicking on: 📖.
• View the definition of some words (in the text or in a “pop-up”).

The student, in addition to the functions explained, can use the Foditic breadcrumbs.

![FODITIC > C2 > index_missions > chap4.1](image1)

Fig 1. 2. Breadcrumb of the document “chap4.1” [2]

Foditic breadcrumb is an improved version of Claroline breadcrumbs because it uses a tree structure (unlike the bread crumbs Claroline, which uses a linear structure). Students can navigate to the next chapter clicking here: 🔄. Foditic breadcrumb does not add a new item in the list, but the last chapter is replaced by the new chapter:

![FODITIC > C2 > index_missions > chap4.2](image2)

Fig 1. 3. Breadcrumb of the document “chap4.2” [2]

like monkeys jumping between branches of the tree. In Claroline, breadcrumb is only available in Claroline pages, not in the documents. In Foditic, they are extended to the documents.
1.2. The Foditic functions intended for course administrators

The Foditic platform includes two stand-alone php programs to automate the technical aspects of the work of the teacher: fodi_import and fodi_export. First, the teacher has to prepare an html document using any html editor. Some predefined html comment tags included in this document with trigger the inclusion of wanted functions in the generated php file.

1. When the teacher wants to give the student the opportunity to add their own privates comments, he can write a html anchor tag whose name must be an different integer for each occurrence:

   \[<A NAME = "1">\] [1]

2. When the teacher wants to show a picture, he can do it writing the next html tag.

   \[<!-fodiImg-->\]

   followed by the <IMG> tag with the URL where to find the picture.

   \[<IMG SRC= "URL">\] [1]

3. When the teacher wants to show his comments, he can do it writing the next html tag and writing his comments between « “ ” ».

   \[<!-fodiCom="<I>my comments</I>"-->\] [1]

   Note: the teacher may use html tags for changing the text format (bold <B>, italic <I>, underline <U>...) in the text of the comment.

4. When the teacher wants to explain a word, he can do it writing the next html tag.

   \[<!-fodiDef="definition of the word","word"-->\] [1]

   Instead of simply 'word'. When the mouse slides over such word the definition is shown.

The differences between the comments and definitions are few. In both cases, the explanation is showed in a pop-up window or in the text.

The differences are, first, if the student chooses the presentation with "pop-ups windows", in the case of the comments pop-up window is activated by clicking on the small blue ball (comm), however, in the case of definitions it is when the mouse slides over such word. On the other hand, if the student choose the presentation with "continuous text", the comments are shown as indented text, while the definitions are shown between " [ ] " in the current paragraph, with a particular font.
5. When the teacher wants to give the student the opportunity to generate a printer-friendly view of a set of web pages, must be add the following HTML comment including a list of documents to include (such as):

```html
<!-fodiPrint = '/DIRCOURS/file11','DIRCOURS/file2'-->
```

NOTE: Any number of files to print, each between single quotes; path starting with / and given from the root of the site.

NOTE: This tag has to add at most once in the head.

6. Another feature added in the stand-alone version of the php program fodi_import, is designed for bilingual courses. It is not added in the function fodi_export nor in the tools of the Foditic document page.

When the teacher wants to show his comments in French, he can do it writing the next html tag, writing your comments between « “ ” ».

```html
<!--fodicom_fr="<I>My comments in French</I>"-->
```

When the teacher wants to show his comments in English, he can do it writing the next html tag, writing your comments between « “ ” ».

```html
<!--fodicom_en="<I>My comments in English</I>"-->
```

If the teacher wants to show his course in both languages, he has to write it with the two tags. So, the student can choose the language.

7. Similarly, the teacher can write plain text in both languages with the next tags.

```html
<!-fodiBLfr="<H1>My plain text in French</H1>"-->
<!-fodiBLen="<H1> My plain text in English</H1>"-->
```

8. When the teacher wants to insert audio material, which is useful in language courses, he can do it by writing the next tag. He has to write 3 arguments:

1. The file URL.
2. If he wants to show subtitles or not (optional).
3. Sequences to let listen (optional).

```html
<!-fodiVoice='mydir/playlist.xml','on or off','1,3-5,7'-->
```
9. When the teacher wants to insert a marker for converting user notes into user answers (editable by the professor to questions of the professor (for language courses mainly)).

\[- fodiQR- \]

After all markers for private notes, marker for inserting a link to "thisfile_solution.php" if all answer are present.

\[- fodiSOL- \] [1]

NOTE: This tag has to add at most once in the head.

Besides, the following changes will be made to the original file:

- All scripts will be moved from the header to the body (in reverse order).
- The html header and footer are replaced with Foditic php headers and footers.
- The fodiPrint php array is created in the php header.
- The Foditic images, comments and definition markers are replaced with the corresponding JavaScript.
- JavaScript for user comments is added after the anchors.

When the html document is ready, the teacher can use the Foditic program 'fodi_import' to transform it into a php program.

If, in the file, there are some link to others html files that will be converted to php, the teacher has to replace himself the .html or .htm in-links by .php.

1.2.1. **Fodi_import**

This program will convert an html file (e.g. mytext.html) into a Foditic php file (called mytext.php).

There are two ways to do it:

- The first way, is from command line:

  \texttt{fodi\_import mytext.html}
The user must previously be logged on the Foditic server and have the necessary permissions.

Or generate the php file on another computer and upload it to the server using a tool outside Claroline (scp).

- The second way is doing as follows, from the Foditic website:

[Assume that the user had already signed up and has logged in and this user is a teacher and the user has navigated to the document page of the web application]. The system responds showing the document page (like in the Claroline version) adding also some links for converting html files into php (one link for each html file).

To provide this function, it was necessary to modify the Claroline file “document.php”. Unlike document.php of Claroline, a new column (called fodi_import) is added to the table and identifies html files. For each one of these files, a link is added in this column for converting the html file to php. (See the preceding figure).

If the user selects one of the links that are associated to an html file, the system returns the same page adding to the URL “?cmd=exFodi_import” and “file=’$cmdFileName’”. With the “cmd=exFodi_import”, the system can detect that the user wants to convert the html file with name “$cmdFileName” to a php file. So, the system creates a new php file with the same name as the original html file ($cmdFileName). If there is an old php file with the same prefix, the system removes it.

To create this php file, the library “fodi_import_lib” for using the function “fodi_import” is imported in the modified document page. The system calls the function “fodi_import” and it notifies that a new document has been successfully converted to php or includes an error.
1.2.2. Fodi_export

This program is the opposite function of fodi_import. The system will convert back a Foditic php file into an html file. The original file XXX.php will be unchanged. The new file will be called XXX.html

This function is only available on the command line

The teacher can call it by writing the following line:

\texttt{fodi\_export mytext.php}

1.2.3. Fodi_gloss

This program will extract \texttt{<!--fodiDef ...-->} records from a set of html files and generate a Foditic php file called glossaire.html. Definitions are sorted alphabetically in this file.

The teacher can call it writing the next line:

\texttt{fodi\_gloss doc1.html doc2.html}

Or doing the following:

[Assume that the user had already signed up and has logged in and this user is a teacher the user has navigated to the document page of the web application]. The system responds showing the document page (like in the Claroline version), adding also the link “Créer un glossaire” (as seen in the above figure).

\textbf{Fig 1. 5. Functions of the page “Documents et Liens” of the Foditic website} [1]
If the user selects this link, the system will show a new page as seen in the next figure. The code of this page is in “/fodi_scripts/fodi_gloss_auto.php”.

![Diagram of files with "fodidef" and selected files]

Fig 1.6. Main page of the “Créer un Glossaire” of the Foditic website [1]

In this page, the user can choose the html or htm files from which he wants to extract the definitions and generate the document “glossaire.html”. In the code of “fodi_gloss_auto.php”, there is a JavaScript allowing user to choose the html files (last picture). So, the user selects the set of files and the system creates the “glossaire.html” with the definition comments of these selected files.
1.3. **Implementation**

I shall now explain how the Foditic functions were implemented and what modifications there were in the Claroline (2005) programs.

1.3.1. **Introduction:**

All the changes made by Foditic are organized in new folders and independent of Claroline files. These folders begin with “fodi_”.

- **fodi_claro_modifs:** In this folder, there are the Claroline files that are modified. These new files must replace a few original Claroline files.
- **fodi_lang:** There are the new system messages (or messages modified) in French of the Claroline platform (it is not the “bilingual” function added by Foditic but data for the instance multilingual support of Claroline programs).
- **fodi_scripts:** Here you will find files with the code for new Foditic features.

1.3.2. **Fodi_claro_modifs:**

1.3.2.1. **Claro_init_footer:**

The *claro_init_footer* function, like the Claroline function, builds the bottom of the pages of the website. In the file “fodi_claro_modifs/claro_init_footer.inc.php” you can see the code of this function.

It uses similar code to the Claroline footer but modifying the text to display (You can see in the follow picture).

- First, is showed the next text:

  “Administrator FODITIC : Foditic Admin”, where *Foditic Admin* is a link for sending an email to the Foditic administrator.
If the user is navigating in a Foditic course page, the system also shows the next text:

“Manager(s) for course: course administrator”, where course is the name of the course and course administrator is the name of the course administrator(s). Additionally, the name of the course administrator(s) is a link for sending an email to the course administrator(s).

- In the right of the footer, there is the next text:

“Powered by Claroline © 2001 – 2005”, where Claroline is a link to the official Claroline website.

- In the center of the footer, there is the next text:

“Avec le soutien du Fonds social européen”. This is a contractual requirement of the FSE that co-funded the development of FODITIC.

1.3.2.2. claro_init_banner:

The claro_init_banner function builds the top of the pages of the website. In the file “fodi_claro_modifs/claro_init_banner.inc.php” you can see the code of this function.

First, is showed the platformBanner. You can see this part in all the scenarios. It shows the next picture:

Where:

- Fondation Chimay-Wartoise is a link to the URL: http://www.cap2010.be/,
- FODITIC is other link to the Foditic main page (http://www.foditic.org/index.php)
- Université catholique de Louvain is other link to the official website of the UCL (http://www.uclouvain.be/).
The header can change depending on where is the user.

- When the user is logged, you can see other part of the footer named userBanner. This part, like Claroline, contains the user section but adding the next text: L’UCL est membre de Cluster, where it is a link to the website http://www.cluster.org.

- When the user is navigating in a Foditic course, he can see other parts of the footer named courseBanner. This part is like Claroline but using a different option list. There are the options: Course Home, Agenda, Announcements, Documents and links, Forums, Groups, Users, Chat and Missions.

Finally, there is the breadcrumb. It is the same as in Claroline but with a few changes, like for example:

- The main page does not show the breadcrumb.
- When the user have selected the “missions” tool of a Foditic course and he wants to navigate to the next chapter, the Foditic breadcrumbs do not add a new item in the list, but the name of the last chapter is changed by that of the new chapter.

1.3.2.3. Document

The document function shows the document page of the Foditic website. The code of this function is located in the file “fodi_claro_modifs/document.php”.

This function is based on the Claroline function but adds a few features.

The first is fodi_import (only is available for the course administrator). The document page includes a new column with a new link for each html file allowing convert this html file into a Foditic php file.

- If the course administrator clicks on one of these links, the system returns the same document page adding to the URL:

  “?cmd=exFodi_import” and “file="$cmdFileName”

With the “cmd=exFodi_import”, the system can detect that the user wants to convert the html file with name “$cmdFileName” in to a php file. So, the system create a new php file with the same name that the original html file ($cmdFileName). If there is an old php file with the same prefix, the system removes it.
To create this php file, the document php page must first import the library “fodi_import_lib” for using the function “fodi_import”. After performing its job, the latter notifies that a new document has been successfully converted to php or that an error occurred.

The second added feature is `fodi_gloss`, that is also only available for the course administrator. The document page shows a new link named “créer un glossaire” allowing to extract <!--fodiDef--> records from a set of html files into a Foditic file called glossaire.html.

If the user selects this link, the system will show a new page as seen in the next figure. The code of this page is in “/fodi_scripts/fodi_gloss_auto.php”.

### 1.3.2.4. Courses:

The `courses` function shows the Foditic page allowing registering in a course. The code of this function is located in the file “fodi_claro_modifs/courses.php”.

This function supports the Foditic policy of cloning each year as new version of each course so that past students still keep access to the old version and new students register to the new course.

This uses the fact that each Claroline course has a real name and a fake name (the fake name is shown by the web server; the real name is used internally). In Foditic, each course identifier ends is a year code (e.g. 10 for 2010). In the variant of the year of a course, this 2 year code is removed in the fake name. Whenever courses are listed, the course of the year are shown first. They are identified by the fact that their fake name differs from the real name.

### 1.3.3. Fodi_lang:

In the folder `fodi_lang` there are two folders more named English and French.

Both folders have a file with the same name: “fodi_comment.inc.lang.php”.

These files there are some variables with the same name but the value of these variables is a text in English or in French.

With this standard Claroline method, when you want to display a message for the Foditic comments, you can write the name of the variable and not a text. So, depending on the language chosen by the user, the system displays text in English or French.

This is standard Claroline policy for displaying messages. This folder includes some Foditic-specific messages.
1.3.4. **Fodi_scripts:**

Here you can find specific Foditic scripts that allow, for instance, students to add and manage your own comments on the documents in the Foditic website.

This folder includes scripts to convert courses-note consisting of html pages into php pages including the Foditic features.

### 1.3.4.1. /imgs:

The folder *imgs* contains all the new images used in the Foditic website.

### 1.3.4.2. Fodi_import:

This stand-alone php program called from the command line generates a Foditic php document from an html document. It proceeds in several steps:

This program will convert an html file into a Foditic php file. The original file XXX.htm or XXX.html will be unchanged. The new file will be called XXX.php

The following Foditic markers may be in the source

1. At most once in the head

```html
<!--fodiPrint = "'/DIRCOURS/file11','DIRCOURS/file2'"-->
```

NOTE: Any number of files to print, each between single quotes; path starting with / and given from the root of the site

```html
<!--fodiPrint = "'/DIRCOURS/file11','DIRCOURS/file2'"-->
```

2. Once in the body

a. marker for converting user notes into user answers (editable by the professor to questions of the professor (for language courses mainly)

```html
<!--fodiQR-->
```

b. (after all markers for private notes, marker for inserting a link to "thisfile_solution.php" if all answers are present

```html
<!--fodiSOL-->
```
3. Any number of times in the body
   a. markers for user notes (private)
      
      \texttt{<A NAME="i">}
      
      Where \texttt{i} is a different integer for each occurrence.
   
b. markers for multimode images
      
      \texttt{<!-fodiImg-->} before the html \texttt{<img>} tag
   
c. markers for multimode comments
      
      \texttt{<!-fodiCom="comment with html tags" -->}
   
d. markers for multimode definitions (for language courses mainly)
      
      \texttt{<!-fodiDef="comment with html tags","word" -->}
   
e. markers for voice inserts; args are file\_name, show text or not, and sequences to show; last 2 args are optional
      
      \texttt{<!-fodiVoice="mydir/playlist.xml","on or off","1,3-5,7"-->}
   
f. markers for text to be displayed only if the user selected language in bilingual documents.
      
      \texttt{<!-fodiBLfr="french text"-->}
      
      \texttt{<!-fodiBLen="english text"-->}

The following changes will be made to the original file:

1. All scripts will be moved from the header to the body (in reverse order)
2. The html header and footer are replaced with Foditic php headers and footers
3. The fodiPrint php array is created in the php header
4. The Foditic images, comments and definition markers are replaced with the corresponding JavaScript.
5. JavaScript for user comments is added after the anchors
1.3.4.3. **Fodi_import_lib:**

This is a version of fodi_import designed to be used as a library in the document page. It is an alternative to the stand-alone php page.

1.3.4.4. **Fodi_export:**

It is the opposite of the php program fodi_import. Converts from the Foditic php document to an html document.

This is a program not used normally by course administrators.

It only can be done it manually (from command line). Then, this function creates a document with the html structure and deletes the students function (for example: add comments, print function...).

NOTE: This program corresponds to an old version of fodi_import. It needs to be updated.

1.3.4.5. **Fodi_bilingual.inc.php:**

The Foditic website has a function allowing the student to select the language to display the contents of the php pages generated by fodi_import, choosing between English and French.

This function saves the language of each user for each subject in the Foditic BD. So, the system can know the language chosen by the user.

Also, the system saves this information in the session to avoid the frequent use of the database.

Finally, the file “fodi_cartouche_BL” is included if the course to show is a bilingual course. It includes the code used to display the explanatory text of the beginning of each page.
1.3.4.6. **Fodi_cartouche:**

When the Foditic system displays a php page including Foditic function, it shows automatically a table or cartridge at the top of php page, like the next picture, explained above.

![Cartridge of a bilingual document](image)

**Fig 1.9. Cartridge of a bilingual document 2. [2]**

1.3.4.7. **Fodi_cartouche_BL:**

This file is same than “fodi_cartouche”, but showing it in the language choosing by the user.

The principle is that it includes both versions but displays the one selected by the student.

1.3.4.8. **Fodi_comment_aide.txt:**

Manual to explain how insert the comments in a Foditic document (with the functions fodi_import and fodi_export). It is not necessary to modify this document.

1.3.4.9. **Fodi_comment_config.php:**

This file includes the definitions of a few variables used by different files in the Foditic scripts.

1.3.4.10. **Fodi_comment.inc.php:**

The function for insert the comments in the Foditic page and the Foditic code.

1.3.4.11. **Fodi_comment_only.inc.php:**

Simplified version of the function “fodi_comment.inc”.

1.3.4.12. **Fodi_comment.js:**

The JavaScript code to display windows, messages to the user guide...
1.3.4.13. **Fodi_comment.php:**

The pop-up window to insert, consult or modify the comments written by the users (like the next picture).

![Fig 1. 10. Window to write the comments. [2]](image)

1.3.4.14. **Fodi_footer.inc.php:**

This is the function for the bottom of the generated page.

This page includes the modified file “claro_init_footer.inc.php” explained above.

If the user wants to create a Foditic printable document, this function also writes the next text to avoid printing this footer in the printed version of the page:

```
"<!-stopprint-->"
"</TD><TR></TABLE>"
```

1.3.4.15. **Fodi_gloss_auto.php:**

The page to build the “glossaire” document. This program will extract <!--fodiDef ...--> records from a set of html files into a file called glossaire.html automatically.

This is called when the course administrator clicks on the link “Créer un glossaire ” in the Foditic page “document and link”.

1.3.4.16. **Fodi_gloss:**

This function generates the Foditic “glossaire” document manually (from command line).
1.3.4.17. *Fodi_gloss.inc.php:*

File to insert the link “Créer un glossaire” in the page “document and link”.

1.3.4.18. *Fodi_qr.inc.php:*

File to insert at the top of the body of the Foditic page the use of "questions and answers" for a practical work (by the students, corrected by the teacher).

1.3.4.19. *Foditic.js:*

The JavaScript accessible by the entire Foditic website.

1.3.4.20. *Mode_emploi_foditic.doc:*

Manual about how to use the functions: fodi_import, fodi_export and fodi_gloss (from command line).
2. Claroline

2.1. About Claroline

Claroline is a groupware (a set of applications that are integrated under a single project and work on many concurrent users for development) and collaborative asynchronous, which allows you to mount virtual educational platforms in seconds and with minimal knowledge both for installation as well as for the administration.

Claroline is a project of Free Software released under the GNU / GPL (GNU General Public License) and is developed entirely by the PHP web programming language, use as a Management System Database (DBMS) the well known MySQL server. Claroline system was initially designed for the platform Free GNU / Linux, but as the system is accessible through a browser or Web browser, it is available in a multiplatform system, i.e. it is possible to install and configure it under any operating system.

Claroline has particular characteristics of CMS (Content Management System), such as being completely dynamic, highly configurable, versatile and simple to modify its contents. Claroline is an ideal system for trainers, teachers and education professionals, which almost instantly will allow them to manage their virtual course e-learning environments and allows us control of the following tasks:

- Publish documents in any format (PDF, HTML, Office, Video...).
- Run public or private discussion forums.
- Manage a list of links.
- Create student groups.
- Compose exercises.
- Structure an agenda with tasks and deadlines.
- Make announcements (also via e-mail).
- Have students submit papers.
- Consult statistics of frequenting and success in the exercises. [3]

To install this platform, you follow the steps explained in the annex (point 5.1: “Claroline installation”).
2.2. Easy handling

The Claroline platform is easy to handle. For using this platform, the teacher only needs to be familiar with his browser. He only has to fill easy forms to create his own courses and preparing locally on his own computer the documents that he wants to upload on the website, without any dedicated technical training.

This way, he only has to work creating his documents for the students and does not waste time on the design and development of the platform. The same way, the students can access to such documents easily.

2.3. Modules

As you know, the platform Claroline is released under Open Source license (GPL). Downloading and using Claroline is then completely free of charge.

If you wish, you can contribute to the project and help the Claroline team in different ways.

A method to contribute to the project is with packages auto-installable named “Modules”. With these modules, you can create new tools to improve the use of Claroline platform, adding new functions for a teacher in a course or adding new administration tools, only accessible by the platform administrator.

Beside, you can to share with the community of Claroline to help the developers to create a better platform.

For installing a module, you can see a video-tutorial about how to install a module in Claroline server:


NOTE: To install the specifics modules “Factory” and “Clone a course”, you can see it in the follow two chapters, at the point 3.1 and point 4.1.

In the follow two chapters, you can see the developer and the use of two new modules of Claroline to adapt the Foditic functions using the Claroline functions.
3. The “Factory” module

Factory is a module of type 'tool' to add several services that are not available in a standard Claroline server.

In Claroline, learning documents can be in any document format, usually html or pdf. Besides these documents, Claroline includes a set of learning tools that use php functions.

The goal of this module is to add new php functions using the Claroline MySQL database to improve the ease of learning the documents including the course material.

They include various services related to the presentation of the documents and allowing each student to annotate documents for himself.

Additionally, they include tools provided to professors to help them in their work, such as generating automatically active documents or cloning the current course. (In order to generate a new version of the course each year and keep old versions accessible to ex-students who learned them.

3.1. Installation of the module Factory

To install the new “Factory” module, you must do the same steps as for the other modules.

2. Login on as platform administrator as explained in point “Login on Claroline” of “Annex”.
3. Go to administrator section, explained in point “Administering the platform” of “Annex”.
4. Click on the link “Modules” in the “Platform” menu to go to the “modules” administration.

![Fig 3.1. Platform administration main page of Claroline](image)
5. Click on “Install module”. Select the checkbox “Package on your computer (zip only)” and click next.

6. Browse through your computer and find the module you have just downloaded.

7. Select the checkbox “Activate module on install”.

8. Select the checkbox “This tool must be activated manually in each course”.

Fig 3.2. List of tools in the main page of Modules before the installation of the “Factory” module

Fig 3.3. Options selected before the installation of the "Factory" module

NOTE: This option is very important, because the system cannot work correctly without it, since if you do not active the module manually, the system may not create the new subfolder, as is explained in point 3.4. “After the installation”.

9. Click on the button “Upload and Install module”.
Now you have installed correctly the Factory module.

To use it, you should to activate it in your course.

### 3.2. Activation of the module Factory

This module can be used in existing courses or new courses.

To activate this module, follow the next steps:

1. Enter in the course where you want to active the module Factory. The first time, the user should not see any link in the “tools list” of a course (the left menu of the course main page).

![List of tools in the main page of Modules after the installation of the “Factory” module](image1)

![Main page of a course](image2)
2. Click on the link “Edit Tool list” to Activate/deactivate a tool in the course main page.

3. Click on the link “Add or remove tools”.

4. Click on ✔️ of Factory to add this module to the tools list.

Now, you should to see the module Factory in the list of “Tools currently in your course”.

You have activated the module Factory in the current course.

3.3. **Change the right of the module Factory**

Claroline allows selecting the tools you want to make visible for the user. An invisible tool will be grayed out on your personal interface. The “Factory” tool is not intended to be visible by students.

The course administrator can also change the access rights for the different user profiles.
To do it, follow the next steps:

1. Enter in the course where you want to change the access rights of the Factory module. The first time, the user should not see any link in the “tools list” of a course (the left menu of the course main page).

2. Click on the link “Edit Tool list”.
3. Click on the link “Manager tools access right”.
   - Click on ☐ to make a tool invisible to the users. ☐ Will appear under Visibility.
   - You can choose the profile who you want to change the rights:

   **Anonymous**: Course visitor (the user has no account on the platform).

   **Guest**: Course visitor (the user has an account on the platform, but is not enrolled in the course)

   **User**: Course member (the user is actually enrolled in the course)

   **Manager**: Course Administrator.

   You can choose between the options “access allowed”, “edition allowed” and “no access”.
3.4. **After the installation**

When the user installs and activates the module, it produces some changes in the system.

In the “tools list” of a course (the left menu of the course main page), the system add a new link: Factory.

- **Factory** is a link to go to the main page of the Factory module (it is explained in the point 3.5: “Main page” of the module Factory). This link is only accessible to the personal allowed. The students have not access to this page.

  NOTE: It is preferable to change this link to invisible mode for everybody except the course manager.

The system also modifies the database of Claroline.

- When the user creates a new course, the system adds a new course table in the Claroline database named “comments”. This table saves the comments added by the users (explained in the point 3.6.1.1 “common tags”).

The system creates another new folder named “Factory” in the course folder. In this folder, the teacher can save all the “private” files. These files cannot be seen by the students.

Also, the system creates a new folder named like the code of the current course inside the folder “Document”.

The teacher must upload all his files about the course in this new folder (exercises, documentation...). He can create new sub-directories inside this folder.
3.5. Main page

The module Factory should not be allowed for the students.

NOTE: The course administrator should to change to invisible mode the link “Factory” from the course tool list.

![Alert when a student tries to access to ’Factory’ module](image)

To access to the main page of the module, you can click on the link “Factory” of the course tool list.

At first appearance, the structure of the main page is like the “Documents and Links”, but there are some differences:

- It shows the files from the new “Factory” folder inside the course repository.

  Also, you can see a new column in the list of the files named “php/html”. This column shows an icon in the files php or html to use the functions claro_import or claro_export, respectively, explained above.

- Besides, there are the same functions as in “Documents and Links” such as upload file, download current directory...

  NOTE: These functions are for the folder “Factory”, i.e., if you use the function “upload file”, the file is saved in the folder “Factory” and not in “Documents”.

- Furthermore, there are three new links to access to the functions “create glossary”, “add entry to course tool list” or “clone this course”, intended to help the course administrator work.
3.6. Functions

The aim of this module is to improve the learner-friendly view of the learning material.

This facilitates the use of learning for students, who can act interactively on the documents provided by the teacher.

Therefore, the module consists of function to help the teacher to produce interactive learning material.

3.6.1. Claro_import

This function will convert the ready html file into a php file. The system will produce a file XXX.php from a file XXX.html

Before using this function, the teacher has to prepare an html document using any html editor. Some of the enhancements will be applied to any html file that is converted: same look and feel as the “Claroline pages” including the extension of the breadcrumbs to the tree of converted documents, including lateral navigation.

I will explain how to prepare the html documents to add extra features, adding special tags.

3.6.1.1. Common tags:

The common tags are special tags that the teacher can in any html document. With these tags the teacher can add the following features:

- When the teacher wants to give the student the opportunity to add their own comments, he can write a html anchor tag whose name must be a different integer for each occurrence:

  `<A NAME = "1"></A>` [4]

- When the teacher wants to let the student choose how to show a picture, he can add the next html tag.

  `<!--fodiImg-->`

  Before the `<IMG>` tag with the URL where to find the picture.

  `<IMG SRC= "URL">`[5]
• When the teacher wants to give the student the opportunity to generate a printer-friendly view of a set of web pages, he must add the following HTML comment including a list of documents to include (such as):

```html
<!--fodiprint="/INGI2346_10/missions/chap2/chap2.1.php",
'/INGI2346_10/missions/chap2/chap2.2.php",
'/INGI2346_10/missions/chap2/chap2.3.php",
'/INGI2346_10/missions/chap2/chap2.4.php",
'/INGI2346_10/missions/chap2/chap2.5.php"
-->[6]
```

• When the teacher wants to explain a word, he can do it writing the next HTML tag.

```html
<!--fodiDef="definition of the word", "word "-->
```

Instead of simply 'word'. When the mouse slides over such word the definition is shown.

### 3.6.1.2. Bilingual tags

The bilingual tags are special tags that the teacher can only write in the bilingual documents. A bilingual document is written for the learning in two languages. The students can choose between seeing the text in English or French.

Experience shows that the best way to maintain a text in two languages is to have the two versions in the same file, interleaving the paragraphs in the two languages.

This way, when the teacher modifies a paragraph in one language, he will naturally update the translation that is in the paragraph just below. This is what the following tags are designed for.

If the teacher wants to make his course available in both languages, he has to write the main text with the two following tags. So, the student can choose the language.

• Teacher can write the main text in both languages with the next tags.

```html
<!--fodiBLfr="<H1>Mon texte en français</H1>"-->
<!--fodiBLen="<H1> My plain text in English</H1>"-->
```

---

[6] [7] [8]
When the teacher wants to show his comments in both languages, he can do it writing the next html tags, writing your comments between « " " ».

\[ <!--fodicom_fr="\<I\>Mon commentains en français\</I>"-->
\[ <!--fodicom_en="\<I\>My comments in English\</I>"-->

3.6.1.3. **Monolingual tags**

The monolingual tags are special tags that the teacher can write only in the monolingual documents or parts of documents. A monolingual text is written for the learning in a single language. The students cannot choose between seeing the text in English or French.

- If the teacher wants to show text plain, he should not add any special tag. He only must follow the HTML rules.
- When the teacher wants to show his comments, he can do it writing the next html tag and writing his comments between « " " ».

\[ <!--fodiCom="\<I\>my comments\</I>"-->

This tag displays the comment without regard for the language selected by the student.

Note: the teacher may use html tags for changing the text format (bold <B>, italic <I>, underline <U>...) in the text included in the tag. This text may even include “”.

When his html document (with or without the preceding tags) is ready, the teacher can convert it into php.

The teacher must go to the main page of the Factory module. He can see all his documents that are saved inside the folder “Factory”, including the HTML documents that he wants to convert to php using the function “claro_import”.


He must click on the icon 🍃 and automatically the system creates, under the HTML file, the corresponding PHP document.

![Figure 3.9: Main page of "Factory" module after to use "claro_import"](image)

This document is saved in the same folder as the HTML document. This way, the user can change between HTML and PHP format changing the extension of the URL in the referring document.

Once the php file is created, the teacher can see it clicking on the name of the php file.

NOTE: The students cannot see this document from Factory, because they are not allowed accessing to that module. They can see it through a tool that can be automatically generated in Factory as explained further below in the point 3.6.4: “Add entry to the course tool list”.

### 3.6.2. Claro_export

This function is the opposite of claro_import. The system will convert back a php document converted by the function “claro_import” into an html file.

The original file XXX.php will be unchanged.

The new file will be called XXX.html.

NOTE: In the case that there is another html file with the same name, the system does not remove it, but the system creates it adding “_2” to the name (e.g. “NAME_2.html”).

The teacher must go on the main page of the module Factory. He can see all his documents that are saved inside the folder “Factory”, including the PHP documents converted by the function claro_import, that he wants to convert to html using the function “claro_export”.

The teacher must click on the icon 🍃 and automatically the system creates, above the PHP file, the HTML document.
Such document is saved in the same folder as the PHP document.

3.6.3. Glossary

This function will extract the tags:

```html
<!–fodiDef=“definition of the word”, “word “–>
```

from a set of html files and generate a HTML file called glossaire.html. Definitions are sorted in this file as shows the follow picture.

![Glossaire de the_burglary](image)

The teacher must go on the main page of the module Factory. He can see all his documents that are saved inside the folder “Factory”, including the HTML documents that he wants to extract the definitions using the function “Glossary”.

![Fig 3. 10. Main page of "Factory" module after to use "claro_export"](image)

![Fig 3. 11. Glossary example](image)
NOTE: The teacher only can extract the definitions of the documents saved in the current folder, i.e., he cannot extract the definitions of the documents saved in the different folders together.

Of course he can change the name of glossary html, and convert it into a php file using claro_import.

The teacher must click on the link  
 to create the glossary. The system shows a page with two tables.

![Fig 3.12. Main page of the "Glossary" function](image)

In the left table, there are the html or htm files from which he wants to extract the definitions and generate the document “glossaire.html”.

In the right table, there are the selected files.

To select a file, he can click on the name of the file that he wants to select from the left table and sliding the mouse to the right table.

He can also do the reverse process. He can deselect the files clicking in the name of the file from the right table and sliding to the left table.

When the teacher has chose the files from which he wants to extract the definitions, he must click the button “Create Glossary” to do it.

Eventually, the system redirect to the Factory main page.
3.6.4. **Add entry to the tool list**

This function is done to allow the students to access to the php files and working with these files. The way to do it is through the “index”.

The principle is to add on entry in the left menu of the main page of the course. This will point to the file index.php of a subfolder of “Factory”.

The teacher must then upload a tree of html files including links corresponding to the structure of the course (e.g. index -> chapters -> theoretical sections ) -> exercises & points

The uploaded html document must then be converted in php documents using claro_import after the html links have been replaced by php links.

The teacher must click on the link [Add entry to course tool list](#) to use this function. The system shows a page with a field to fill and a button.

![Fig 3. 13. Main page of the "Add entry to course tool list" function](#)

The teacher must fill it with the name of the link that he wants to show.

When the teacher click on the button, the system creates:

**The folder:**

The system creates a new subfolder in the “Factory” folder. The name of the new subfolder is like the field filled in the form by the teacher.

**NOTE:** The name of the new subfolder and the field filled by the teacher may be different, because the name of the subfolder may have dangerous chars that should be replaced for the system to work correctly.
NOTE: Inside the new folder there should be a file “index.php”, created with the function “claro_import”, from an html file “index.html” uploaded by the teacher.

This file contains some links to php files created by teacher with the function "claro_import".

The link:

The system creates a new link in the tool list of a course page (the left menu of a course main page). This link is named like the field filled by the teacher. The path of this link is to the index.php saved inside the subfolder with the same name (…/Factory/NAMESUBFOLDER/index.php).

3.6.5. Clone this course

The aim of this function is to create a clone of the current course. This way, the teacher can create a new course for a new year or simply create a backup of the current course in the Claroline server.

The cloned course has the follow features:

- A copy of the files from the current course. Of this way, the teacher will be able work with the same files than the current course, without the need to upload again the same files.
- A copy of all the information about the current course. It keeps the name and the email of the titular that made the course, the faculty, the language, the visibility of the course...
- Keep the mode of the enrolment to the students of the current course. New students only will be able to enroll in the new course if they can enroll in the current course, but the teacher can change this immediately.
- No students are registered in the clone course.

The system creates a new folder with a copy of the current course and the teacher is asked what code to give to the clone.

For instance if he wants a version of the course for each academic year, he should select codes ending with a year id, say C1_10 for its course used in 2010. Then he should use the name C1_11 for the course used in 2011.

The teacher must click on the link [Clone this course](#) to use this function. The system shows the main page to clone the current course.

To clone this course, the teacher has to follow the next steps:
**Step 1: Download**

In this step, the teacher can download a zip file with all the content of the current course that will be used in the clone.

To do it, he must fill the next fields of the form:

- **Path to download the file on your computer:** The path of the folder where the teacher wants to download the zip file.
- **Name of the file to download:** The name of the zip file that the system will create in the chosen path.

Once the form filled, the teacher must click on the button “Download Archive” to start to download the zip file.

If he does not want to download the file, he has the possibility of skip this step clicking on "Skip download". Then, the system will go directly to the step 3 (Clone).

NOTE: if the chosen path does not exist, the system will create it.
**Step 2: Upload**

In this step, the teacher can upload the zip file to the current Claroline server.

The system shows the three fields in the form:

- The path of the server in which to upload: it is the path of the current server of Claroline.
- The path where the file will be uploaded on the server: The folder where the zip file selected will be uploaded.

NOTE: the previous two fields are may not be modified. They are only information for the teacher of where the file will be uploaded.

- Select the file to upload: Click on the button to search in your own computer the file he wants upload.

Once the file to upload selected, one must click on the button “Upload Archive” to upload the zip file to the current server of Claroline.

NOTE: The reasons to download you course on your own computer are twofold:

- It allows archiving your course on your computer and not only on a remote server.
- It allows duplicating (clone) its course on a different server than the one where the original course is located.
Step 3: Clone

In this step, the teacher can clone the current course.

Before cloning, he can also choose the information about the new course before of clone it. Also, he can change the administrative number of the old course.

The administrative number is the code to show to the students in the home page of Claroline.

NOTE: By default, the administrative number is the same as the code of the course.

He can:

- See the code of the current course in the field of the form "Name of the old course".
- Change the administrative number of the current course in the field of the form "New administrative number of the old course".
- Change the code of the clone in the field of the form "Name of the new course".
- Change the administrative number of the clone in the field of the form "Administrative number of the new course".

Once chose the information, the teacher can clone the current course clicking on the button "Clone".
NOTE: The teacher cannot choose the code of the new course if another course with the same code already exists.

NOTE: Can be twice the same administrative number.

*Step 4: Success*

In this last step, the system shows that the course has been cloned successfully.

![Fig 3. 17. Step 4 of the "Clone this course" function](image)

Click on the link “continue” to go to the Factory main page.
4. The “Clone a course” module

Clone a course is a module of type ‘admin’. This module is only accessible by the platform administrator.

The main goal is allow to the platform administrator the possibility of clone a course. He can see all the courses created in the Claroline server and choosing one to clone.

Of this way, the platform administrator can create a new course for a new year and additionally, he can access to a backup about the current course in the Claroline server.

The cloned course has the following features:

- A copy of the files from the selected course. Of this way, the titular of such course will be able work with the same files than the selected course, without the need to upload again the same files.
- A copy of all the information about the selected course. It keeps the name and the email of the titular that made the course, the faculty, the language, the visibility of the course...
- Keep the mode of the enrolment to the students of the current course. New students only will be able to enroll in the new course if they can enroll in the current course, but the teacher can change this immediately.
- No students are registered to the clone course.

The system creates a new folder with a copy of the current course and the teacher is asked what code to give to the clone.

For instance if he wants a version of the course for each academic year, he should select codes ending with a year id, say C1_10 for its course used in 2010. Then he should use the name C1_11 for the course used in 2011.
4.1. Installation of the “Clone a course” module

To install the new module “Clone a course”, you must do the same steps as for the other modules.

2. Login on as administrator platform, explained in point “Login on Claroline” of “Annex”.
3. Go to administrator section, explained in point “Administering the platform” of “Annex”.
4. Click on the link “Modules” in the “Platform” menu to go to the “modules” administration.
5. Click on “Install module”. Select the checkbox “Package on your computer (zip only)” and click next.
6. Browse through your computer and find the module you have just downloaded.
7. Select the checkbox “Activate module on install”.
8. Click on the button “Upload and Install module”.

Now you have installed correctly the module Clone a course.

4.2. Main page

This module is only available to the platform administrator.

If you want to access to this module, you have to log in as such (explained in point “Login on Claroline” of “Annex”).

In the administrator section, you can see a new link named “Clone a course” to clone a course.

It proceeds in several steps:

Step 1: Select a course
In this step, the platform administrator has to choose the course that he wants to clone.

He can see all the courses created in the Claroline server in the list.

Once the course selected, he must click on the button “Select”

**Step 2: Download**

![Image of Claroline interface showing step 2 of the "Clone a course" module]

In this step, the platform administrator can download a zip file with all the content of the current course that will be used in the clone.

To do it, he must fill the next fields of the form:

- **Path to download the file on your computer**: The path of the folder where the platform administrator wants to download the zip file.
- **Name of the file to download**: The name of the zip file that the system will create in the chosen path.

Once the form filled, the platform administrator must click on the button “Download Archive” to start to download the zip file.

If he does not want to download the file, he has the possibility of skip this step clicking on "Skip download". Then, the system will go directly to the step 4 (Clone).

**NOTE**: if the chosen path does not exist, the system will create it.
Step 3: Upload

In this step, the platform administrator can upload the zip file to the selected Claroline server.

The system shows the three fields in the form:

- The path of the server in which to upload: it is the path of the current server of Claroline.
- The path where the file will be uploaded on the server: The folder where the zip file selected will be uploaded.

NOTE: the previous two fields are may not be modified. They are only information for the platform administrator of where the file will be uploaded.

- Select the file to upload: Click on the button to search in your own computer the file he wants upload.

Once the file to upload selected, one must click on the button “Upload Archive” to upload the zip file to the current server of Claroline.

NOTE: The reasons to download you course on your own computer are twofold:

- It allows archiving you course on your computer and not may on a remote server.
- It allows duplicating (clone) the course on a different server than the one where the original course is located.
Step 4: Clone

Fig 4. 7. Step 4 of the "Clone a course" module

In this step, the platform administrator can clone the selected course.

Before cloning, he can also choose the information about the new course before of clone it. Also, he can change the administrative number of the old course.

The administrative number is the code to show to the students in the home page of Claroline.

NOTE: By default, the administrative number is the same as the code of the course.

He can:

- See the code of the selected course in the field of the form "Name of the old course".
- Change the administrative number of the selected course in the field of the form "New administrative number of the old course".
- Change the code of the clone in the field of the form "Name of the new course".
- Change the administrative number of the clone in the field of the form "Administrative number of the new course".

Once chose the information, the platform administrator can clone the selected course clicking on the button "Clone".

NOTE: The platform administrator cannot choose the code of the new course if already exists another course with the same code already exists.

NOTE: Can be twice the same administrative number.
Step 5: Success

In this last step, the system shows that the course has been cloned successfully.

Click on the link “continue” to go to the Claroline main page.

NOTE: A difference of the function “Clone this course” of the “Factory” module, when you clone a course, copy the uploaded zip archive in the folder of the new course. You must to take careful with this step, because you can put the system in risk if you upload a dangerous zip file. Beside, if the chosen upload file is not a zip file, the system copies the folder of the last course in the new course (as do the function “Clone this course” of “Factory” module).

NOTE: If you want to clone a course on a server B, with an archive obtained from a server A, you must on B:

- Create a new dummy course.
- Start the cloning procedure with it.
- Upload the zip archive coming from server A.
- Clone: you get on B a clone of the course existing on A.
- Delete the dummy course.
5. Conclusion

The current work has been very rewarding because I have discovered new learning platforms and the use of these. In addition, I have improved the ability to implement new features using predefined functions provided by other platforms for improving these.

On the other hand, I have expanded my knowledge in the use of a system based on LAMP: Linux, the Apache server, the manager of MySQL database and PHP scripting language.

As for Claroline, I advise all those teachers who think that is more important the management than the container of the learning material itself, since it is a fantastic platform for eLearning usable.

In addition, Claroline has its official website, www.claroline.net, where we can find a lot of information both since the point of view of a user, finding everything you need to initiate us in this excellent platform, or as a point of view of a developer.

Furthermore, it is also intended for those people who want a fast system based on LAMP. Thus we get a system that prevents lockups, which is powerful and fast for many users. In fact Claroline is very powerful and fast, since supports and runs many courses and users efficiently.

I think that the interest on this platform throughout the world has been very large. This has been the basis for write many articles and papers about that platform.

On point of view of a Claroline developer, there is a great and extensive library in which you can find most of the necessary actions, all their functions explained in detail, helping anybody to understand easily the implementations.

The only drawback I have seen is the choice of the platform structure. In my case, the idea was kept in a folder, other than "document", files only to the teacher, without that students could view the contents of that folder. I think we should have the option to change this detail, or when you install the platform on the server, or using parameters in the functions of Claroline libraries.

As Foditic I think it is an excellent website for learning because it provides great new features to improve the system of student learning.
Thanks to new Foditic functions, users are able to study documents, attached by the teacher, more easily and effectively, with the ability to interact with these documents. This is because, it should be better and easier, study using (in the own web site) notes, where students can write or express their clarifications in different parts of the document, see the comments written by the teacher, get definitions of some words just sliding the mouse over them, and so on.

Regarding the creation of these new documents, I think that thanks to the implementation of the website, the creators of a course can use the functionalities with ease just by typing special Foditic tags in their html written documents.

The disadvantage of Foditic, is that there are some php programs that are only able to run by shell, without the use of graphical interface, which many teachers are not able to use these because of their complexity since, there are many teachers who do not know enough information to perform these functions.

On the other hand, one of the great advantages of the functions of Foditic is that they are well commented, explaining the use of each function in detail.

To work these functions, Foditic developers have had to modify the code base of the Claroline platform, such that I think it is a disadvantage to Claroline, since despite being a great open source platform, there are a number of restrictions that prevent incorporate new functions to the platform.

In summary, create these two new modules for Claroline platform is a great advantage for all those who want to use the Foditic functions and I hope it will be useful and efficient for teachers and students that will make use of these improvements.
REFERENCES

Reference [1]


Reference [2]

http://www.foditic.org, 2006

Reference [3]

ANNEX

1. Claroline installation

1.1. Before installation

To install Claroline 1.9X on a computer, whether locally or remotely, you need the following on your machine:

1.1.1. Operating systems

You need one of the following Operating systems:

- Linux / BSD / Unix (*)
- Windows (9x, Me, NT4, 2000, 2003, XP)
- MacOS X.

*Note: Claroline have been tested on Red Hat, Debian and Ubuntu GNU/Linux distributions, but other UNIXes and Linux distributions should work too.

It has received positive feedbacks from users working on Mandriva and Suse. It also heard of user managing to run Claroline on a Solaris box.

1.1.2. Web server

They recommend Apache, but others should work too:

- Apache 1.3 or 2.0
- Microsoft IIS

1.1.3. PHP

PHP scripting language (version 5.1 or later), configured with the following modules: MySQL, zlib, preg. Most of the PHP5 compiled distributions are provided with these modules. Anyway, their presence is checked by the Claroline install script.

Some users could meet problems if their PHP setting doesn't fit these ones:

- safe_mode = Off
- magic_quotes_runtime = Off
- short_open_tags = Off
1.1.4. MySQL databases server

The MySQL databases server (version 4.23 or later) plus a login/password allowing administrating and creating at least one database.

Claroline can be installed in two different modes: 'Single Database' and 'Multi Database'. In 'Multi Database' mode, Claroline will create a new database for each course created. This mode is sometimes more appropriate to manage a platform with a huge bulk of course sites. But it means the database account used by Claroline has to allow creating and administrating several databases. By default Claroline is set to be installing on a 'single database host'.

If you want to use Claroline in 'Multi Database' mode, you have to select this option on purpose at install.

WARNING:

- If you use the single database install option, it is recommended to use this database for Claroline use only. If you share this database with other applications, interferences could occur between table names of both applications that could result in serious problems.

NOTE:

- The backticks characters (``) inserted inside most of the Claroline SQL queries since Claroline 1.3 doesn't work with MySQL versions previous to 3.23.6.

IMPORTANT NOTES:

- Claroline works with only one server and only one account.
- Claroline needs at least one database.
- With installation option single database, Claroline is hack to be able to work with an account which can't create databases.
- With installation option multi databases, Claroline is build with the idea that the account can create databases.
1.1.5. Rights on folders

Give write access to web directory where Claroline has been moved or unzipped to. Remotely, you need to be allowed to change rights permissions on folders and files through FTP, telnet, ssh or any means.

If you don't want to set write access on the whole folders, which are recommended for security reasons, give to the web server user write access on these folders:

With Claroline 1.9.x:

- / (root directory)
- courses
- platform
- module
- tmp

1.1.6. Mail server

A mail server as called Mail Transport Agent (MTA).

This is not absolutely necessary, but some Claroline features will remain silent if you don't provide MTA.

- Most GNU/Linux distributions provide MTA (Sendmail, Exim, Qmail, PostFix, ...).
- On MS Windows machines, no MTA are provided by default. You can either modify the configuration file php.ini to redirect mail to a MTA available on another machine, or install a MTA on the Claroline server. Several MTA are available for MS Windows (Blat, Netmail95, WMailto, Hamster ...).

1.1.7. MySQL requirements

1.1.7.1. Tables repository

Claroline needs 2 types of tables repository (in one or several databases):

- Main tables
- Course tables
1.1.7.2. Main tables

There are tables:

- Central (users, tools list, ...)
- Statistics of the platform

These tables are created in one database.

1.1.7.3. Course table

There are tables for:

- Course tools
- Groups
- Statistics of the course

Claroline is built to have one table for each "course-tool" pair. So table for chat in course foo is not the same table for course bar.

This means that there will be many tables. It's more user-friendly to have one database for each course but before install, they don't know how many courses would be created. So Claroline needs a user with database creation rights (its multi databases option at the installation).

A lot of MySQL administrators would find it crazy: 'haw, give db creation rights to my users ... NO!'

If the administrator still holds on their position, you can request one database and shake all tables of courses in the same database (it's the single database option at the installation).

**Note to the administrators**

A very good solution is to request a right to create databases with a fixed prefix:

- Create a user with less right as possible (nothing)
- Give all access for this user on db 'foo%

With these rights, the account can create all databases he needs but all these databases must begin with foo name. You can set this database name prefix at the installation.
Note to the developers

The solution for administrators is also great if you need to run many Claroline platforms on the same machine. Personally they have a user called "Claroline" with a password. The rights for this user are "all" on "claro%" database. And when they install a Claroline, they set as prefix "clarofoo" where "foo" is their checkout name.

1.2. Prepare the installation

1. Download 'claroline19x.tar.gz' (UNIX, Linux) or 'claroline19x.zip' (Windows).
2. Open a shell, go to download directory and type:

   ```
   tar -zxvf claroline19x.tar.gz
   ```
   
   Windows: unzip 'claroline19x.zip' with your favorite application.

3. FTP or move by any means 'index.php' file and 'Claroline' directory to your website on web server (whether web root or subdirectory). The file 'index.php' should remain where it is, outside 'Claroline' directory.

   **UNIX and Linux**

   Copy the claroline19x directory to the document root of your web server (typically 
   /var/www/).

   **Windows with wampserver**

   Drag and drop the claroline19x directory to the wampserver document root folder, typically 
   c:\Program Files\wamp\www\n
   **MacOSX with MAMP**

   Drag and drop the claroline19x directory to the MAMP document root folder 
   /Applications/MAMP/htdocs/

4. The following folders need to be readable, writeable and executable for everyone:
   - claroline19x/
   - claroline19x/module and all its subfolders

   You can change the access rights from the command line under UNIX or Linux:
chmod ugo+w claroline19x/

chmod -R ugo+w claroline19x/module

5. Open Web Browser and go to :

http://www.mydomain.org/mysite/mydir/claroline/install/

Windows with wampserver locally

http://localhost/claroline/install/ (don't forget the final /).

MacOSX : with MAMP locally

http://localhost:8888/claroline/install/ (don't forget the final /).

6. Follow instructions.

Windows

If you use wampserver out of the box, default connection parameters for MySQL are:

login: root

no password

MacOSX

If you use MAMP, default connection parameters for MySQL are:

login : root

password : root
1.3. The Claroline installer

**Step 1 of 9: Installation language**

This step tries the language of the installation.

**Step 2 of 9: License**
Step 3 of 9: Requirements

This step tries to check some of your config values.

Some values should be changed to have a working Claroline.

Step 4 of 9: MySQL Database Settings
Mysql connection parameters

Enter there the parameters given by your database server administrator:

- **Database Host**: e.g. localhost
- **Database Username**: e.g. root
- **Database Password**: e.g. STQRnbUt

Database usage

In multi database mode, Claroline installs one central database and a new database for each course created. Each course has its own database. Loading speed side, there is no difference between single and multi db. Actually, the system creates the same number of tables in single db mode than in multi db mode. It simply aggregates them in a single database.

So, usability side, it means it's a bit more complicated to manage on a daily basis -- at least with the tool they used (phpMyAdmin). Single mode is used especially if MySQL manages other db or if you are hosted by a provider who leaves you access on a single db.

- **Database mode**:
  - Single
  - Multi (one new database created at each course creation)

**Step 5 of 9: MySQL Database and Table Names**

![MySQL Database and Table Names](image)

The default name of the main tables is “cl_” + name of the table.

The default name of the course tables is “c_” + name of the table.
Step 6 of 9: Administrator Account

This step you can fill the fields of the account for the Administrator user. This user will have all the privileges to the Claroline platform.

Step 7 of 9: Platform settings

You can fill the next fields:

Campus:
- **Name**: Indicate here the campus name, it should be visible on each page header of the campus.
- **Absolute URL**: Verify if URL is correct and corresponding to your server.
- **Path to courses repository**: The path of where you want to save the courses.
- **Main language**: It's the language by default The teacher will be able to change it at the creation of the course.
User self-registration:

- **For Simple user:**
  - *Enabled:* Enables everybody to self-registrate in campus as user, and to consult any courses that are not public.
  - *Disabled:* Doesn't allow anybody to self-registrate in campus as user, but administrator or teacher will have to encode each user.

- **For Course creator:**
  - *Enabled:* Enables everybody to self-registrate in campus as teacher, and to create courses.
  - *Disabled:* Doesn't allow anybody to self-registrate in campus as teacher, but administrator will have to encode each teacher.

User password storage:

- *Clear text:* Password will be visible in database. Administrators are able to consult it.
- *Crypted:* Password is crypted in database.

Notes:

After installation, it’s possible to modify these parameters in the file:

*Claroline/inc/conf/claro_main_conf.inc.php.*

**Step 8 of 9: Additional Information**
Step 9 of 9: Last check before install

Claroline setup successful

Claroline setup successful
2. The Campus website

2.1. Home Page

On the Campus home page, you can:

- Self-register to be able to log in to the campus (if your institution authorizes it).
- Log in to your personal home page with your personal course list.
- View the list of open access courses. To view restricted access courses, you must be logged in to the campus.
- Request your username and password if you forgot them.
- You can search a course on the platform with the search option.

2.2. Login on Claroline

Each user (student, teacher and administrator) may have a profile on Claroline. Each profile is linked to a username (also called login) and a password.

'Log in' means filling in the login form on the campus home page to be authenticated as the owner of a profile.

- Fill in the **Username** and **Password** fields in the form.
- Click on the **Enter** button.
For installing a module, you must log in as a “platform administrator”. To enter on the platform administration page:

- Insert your administrator Username and Password.
- Click on Platform Administration at the top of your home page.

The administrator Username and Password is the profile registered in the Step 6 of 9: Administrator Account of the Claroline installer.

Also you can edit it with a tool in admin/managing/addAdminInhtpassword.php

2.3. Administering the platform

Once logged as a platform administrator, you can access to the platform administration clicking on the link “Platform administration” in the Campus Home Page.
2.4. Platform configuration

This is the main page where you can change the configuration of...

...the whole platform

...the courses

...the users

...the course tools

...the authentication

...the groups

...the kernel
3. Modules of Claroline

With the modules system, you can build your own Claroline extension.

- Create or install your own tools for Claroline and share them with the community.
- Personalize your platform tools list by activating or deactivating existing tools, for all the courses at once.
- Personalize the existing banners and menu in Claroline by adding your own contents or links.

There are some examples of modules that you can find in the website of Claroline, as:

- "Who is online"
- A survey tool to create and use surveys in your courses
- A complete search engine based on php, using xml based indexation and metadata
- And lots of other tools to come.

3.1. Types of extensions

There are 3 types of extensions:

Tools:

Tools that can be added to your courses. Use the module tool in platform administration to install modules. A course module is by default installed in all courses after you activate it but it remains invisible until the administrator or the course administration set it as visible.

Applets:

Applets are components that can be added in user interface to add functionalities. Use the module tool in platform administration to install applets.

Icon sets:

Icon sets replace the original icons of Claroline to change its look.

* Information extracted from the written documentation by Claroline team, 2010.