OCTOPUS RESOURCE CENTRE
REQUIREMENTS AND TECHNICAL SOLUTIONS

João Carlos CHOURIÇO*, Luís Carlos BRUNO**
*Universidade de Évora, Évora/Portugal - Largo Conde de Vila Flor-7000 Évora - jcarlos@dpe.uevora.pt
** ESTIG, Beja/Portugal - Rua Afonso III, 1 - 7800-050 Beja - lbruno@estig.ipbeja.pt

Abstract: Octopus Project is the result of the idea of creation an on-line resources centre, destined for the environmental education. Its implementation was carried through the resource of diverse technologies, which gave body to the initial idea. Among these we must refer the fact that this Resource Centre would not only keep the register of existing resources in diverse places, but also will store its own resources. The Project is internationalised, and was based in the use and development of several tools and moods that respect this characteristic. During development of the Octopus, a strategy of sharing administration was adopted to allow fast updates of the information, as well as its translation. Several tools were developed that allowed to support this management approach assuring the integrity control of existing information. Among these tools we can refer the menus management system and the document editor. In order to assure the resource construction, oriented to the accomplishment of learning tasks, the document editor was improved according to those needs. The on-line Octopus Resource Centre can be accessed in the Internet address www.octopus-eu.org.

Key words: *Octopus solutions, Web technology, online html editor*

I. PROJECT GUIDELINES

The Octopus project emerged from the development of the idea of building an on-line resource centre aimed to contribute for environmental education. In the initial idea was the aim to solve some structural problems capable to supply some of the deficiencies that are known into other on-line resources centres.

In beginning it was established that these differences would allow a significant gain in the process of uploading the information tasks by generality of users, as well the manner as the information management would be made.

One of the base concerns, that was an aim to the development of the project, was based in the idea that Octopus Resource Centre, wouldn't have just lists or repositories of information, for other web sites, or references to external resources, like cd-roms or bibliographical lists. For this system was required that it would allow creating and hosting its own resources. To achieve this, we needed to develop a set of submission management tools that could work with different kind of resource types like texts, images, sounds and videos giving the Resource Centre an effective multimedia capacity. The properties that allow the registration of self-resources, are the same that also allows to make the registration, if needed of external resources, like web sites or the mentioned bibliographic lists.

Our initial idea for a Resource centre didn't end in the hosting of the resources. The concept was extended to the aspects relating with change of information between users. The development of these characteristics, would allow having the users informed of activities, or of the existence of specific and updated resources **[1]**. The connection by Internet with the existence of communication tools would increase the possibilities of each user act as a human resource to the others. The project should be developed in six different languages: English, of common use and support to project, Portuguese, Spanish, Romanian, Italian and Greek that correspond to the diversity of the origin of the different involved partners. This development must assure that all these languages would work simultaneously which was a defying task and a challenge to the projects design.

One another type of concern was about the way information management and of resources centre it would be done. The distance between partners and the need of the information to be placed in the Centre, would suppose a good usability in the use of tools for doing its submission and an immediate access to the information by users. Also any registered user would be capable of operate the system, without needing deep technical knowledge.

The management structure should be light and powerful, in order to allow this type of uses and capable to provide the users to carry simple tasks as to add, modify or delete resources. This way we could assure that the growth of the information of the Octopus, never would be limited of its technological and technical, but only of the will and capacity of its users. This resource centre was based in an "open ended" environment, where its growth, development and continuity were assured, after the term of project that it allowed to create. This "open ended" approach had a double perspective. The related one with the centre and its growth, but also with the learning aspects, since we can't forget that the Octopus would have to be one contributor for the accomplishment of learning, in an autonomous way or not, in the field of environmental Education. We can say that, in this "Open Ended" perspective, the resources centre would have to assure that have the tools and mechanisms to its proper expansion, in terms of new resources and options of use; in educative terms, it would have to allow, that through its use and exploration the users

could make new learning and acquire new capacities and abilities, adding them to those that already have.

II. MAIN DECISIONS

The main decisions taken in the beginning of technical development of Octopus Project can be identified in three different perspectives: *the organization and display of information; the way as options work and are integrated with other and at last, the basic technological structure to use*, that allow to assure, the first two could be assumed.

Organization and Information display

After several tries, we adopted an information organization that reflected also in the information display. We arrived to a screen layout, where the spaces are supposed to be different rooms, where each kind of information is grouped. Like this some basic "rooms" were defined as *Reception*, room related to the attendance of the users; *to Learn*, room where users can access to Octopus resources; *Lab*, where the users with high level permissions can build and change the information related with the resources; *Search*, where the different ways of making the search were grouped.

These were the definitions where the entire project was based. All the rest, are options that emerged with it development and that exists actually, are examples of the way Octopus Resource Centre works and of its possibilities of expansion, although don't represent the same kind of basic structure as the others.

In spite of this kind of organization, some of the options, considered as more important, could be more hidden than expected or desired, with a bigger difficult to access them. For this reason, it was necessary to bring alternatives with bigger visibility. Te left side of the screen was used for that, allowing the system to put there the desired options.

The adoption of this kind of organization, allowed to assure, a great flexibility and simultaneously the growing of the Resources Centre and of the options displayed to the users.

Working Definitions

To assure the working of the Resource Centre, in simultaneous with different languages, related with the needs of quick updates in all of them, the administrative structure couldn't be high hierarchic and with several levels of decision.

To avoid this problem, was decided that Octopus would have a shared administration, considering each partner of the project as an administrator. This could allow, a quicker updating of information.

At language working, this kind of administration could bring some problems, related with a lack of information integrity, if some administrator puts in the Resource Centre, documents not understandable for everybody. To avoid that was decided to maintain the use of the English as the basic language, every time a document, or a new option would be charged for the first time. This way all the languages would have the same menus and options.

The first publication of the menus would be initially in English, giving some time to the partners to translate it to native language, and assuring that orphan links wouldn't exist. At the moment the native language is available the English version would be replaced [**2**]. This way of working, also allowed having specific options, just in one language, if a partner needed to specific aims or events.

It was our permanent concern, related with the characteristics of the shared administration that integrity of information would be assured, also because of the using of different tools and working uses in creating and editing information. This could bring to the users different formats and styles that would confuse the users and giving them a uncomfortable use of the site. For this reason, a group of options were defined that have the result in the development of a set of tools needed to generate the information needed to Octopus, resulting this in a way of standardize the methodologies. This way the initial registration and next editions, needed to make, by the adoption of a unique system of registration, also needed to make the searching work.

The submission of the information to Octopus, this way, would obey to an adoption of a unique tool, an editor with characteristics and working like "Microsoft Word, with large diffusion and very known by most of users.

Many aspects that are basic to Octopus development are supported in a cooperative way of work, needed to assure its growing and development, for what more people and resources are needed.

Based in this idea, Octopus allows the registration and identification of the users, so these could be invited to join in this process of growing. For that different users profiles were defined, so every administrator could change the user initial registration profile, to an other of higher permissions, allowing some users to do more tasks, like creation and submission of resources and transferring to them more responsibilities.

The Octopus Resource Centre, don't have restriction in using and exploring by the unregistered users. We wanted to have means and mechanisms to allow that information consuming users could be more participative and become creators of information in a cooperative philosophy of work **[3]**.

III. MAIN TECHNOLOGICAL DECISIONS

Server side

After analysing the specifications of the system main requests, the technologies to use, that could bring to us best results for the project development were chosen. Based in the identification work and of the main features to implement, in the kind of services to bring to users, in the degree of needed interaction and in the dynamic generation of contents, our option was made to the platform LAMP (Linux, Apache, MySql, Php), as the technological base to develop the Octopus project.

This platform that works by the server side, is supported by Linux [4], an operating system based in Unix [5], which code is freely distributed to general public (GNU – General Public Licence). This allows programmers develop computers programs without to pay software licences and making their own code available for other programmers.

Because Octopus needs other different kind of services, as mailing lists, a control version system, or the online conversion of Microsoft Word [6] files to HTML [7], we concluded that Linux, was the best operating system to assure all the needed requirements. This decision was made based in its robustness, flexibility and scalability, also as the lowest costs.

One of the most successful software based in the philosophy of public programming license is the http Apache web server of "The Apache Software Foundation" [8]. It allows responding to the browser requests, when a user demands to see an Internet page, using the communication protocol http (hypertext transfer protocol). This protocol has a specific language that assures the connection and change of information between server and clients in Web environment. The Apache server is the most used in the entire world, according to an all world study from the Netcraft entity [9], with 64% of web site all over the world using this software. Some security mechanisms are also assured by the using of this server.

Having Octopus Resource centre a great dynamic, in the management of all the information, related with menus, graphics, users administration and others, all the information must be registered in a non-volatile database, in a data structure normalized and systematized. It was needed to make options and choose a relational database that could manage all that information. The chosen system to do that was MySql [10], also a public licence software project, having all the characteristics needed as easiness of managing using, and security in maintain data integrity and accessed protection. A decisive in choosing this software was the use of SQL language (Structured Query Language), a normalized way of work of informatics community, with its simplicity in achieving the data in databases.

The Octopus system must have intelligent mechanisms that respond to users interactions and their functional and informational dynamics. With these demands the HTML, doesn't have the power to do it, because it just describes the displaying of the information and doesn't do its manipulation. To solve these problems, was decided to use the PHP (Hypertext Preprocessor) language [11], as the tool for the computer programming. This language, also a public licence one, can be used for different purposes and is good for web developing uses.

The PHP code has, real programming resources, as others, and is organized in scripts (sets of lines with informatics instructions), which are inside HTML code. This way we can congregate the presentation of the information, by HTML, with the functional way of the system, generated by PHP. This language has, also mechanisms of connecting to database manager MySQL and operate over the file server system, which allows the management of contents that are sent to the users.

In a graphic way we can show in Figure 1, the functional architecture, from server side, of Octopus Resource Centre.

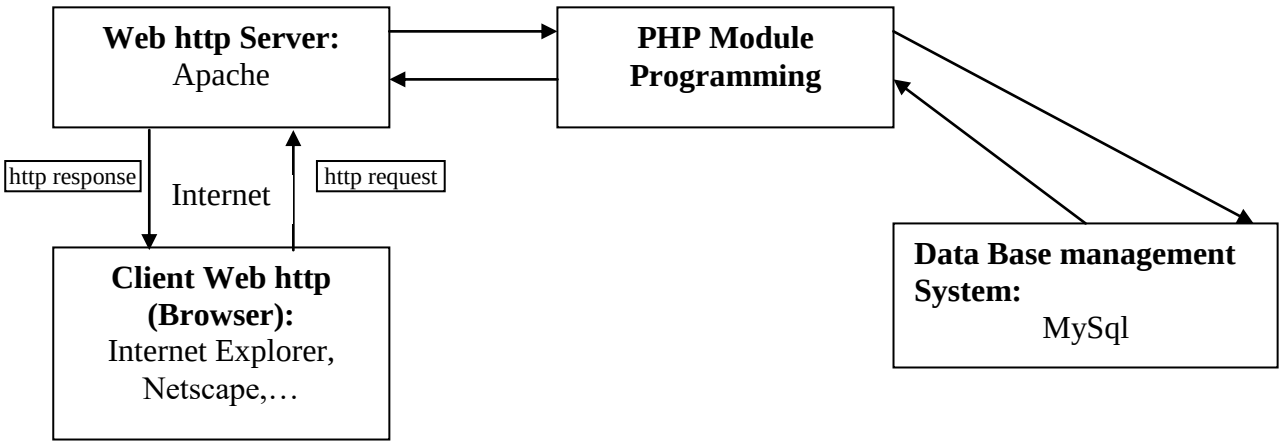

Figure 1– Octopus Functional Web Architecture

Based in this model, we can verify that requests from the browsers to the web server, to open a static content, for instance, an image, or to save the data in a database, are made trough a http message, received by the Apache Server. In the first case, the server answer to the browser requests, sending the content of server file system. In the second case, if any request demands for an execution of a PHP script with access to database, the Apache server send it to the module of interpretation and execution of PHP, which generates dynamically a content received from database.

After the format of answer content, the data are sent between the architecture components, in the reverse order they were made.

Client Side

In a Web environment, the processing tasks of information can be done by server side or from client side (browser). The tasks from client side is more volatile (don't save the information) and are based in the principle that must free the server from additional processing and avowing more loss of time in internet communication. As so, the Javascript language [12] and Java [13], were used to support the local processing tasks and the user interaction with graphic objects from interface. Also CSS (Cascading Style Sheet) [14], was used to standardize, the appearance of documents.

The Javascript programming language is developed by Netscape, Inc. [15], and it's based on embedded scripts in HTML documents. They add them capabilities of doing calculations and interaction with users (management of events) and with other modules of the browsers. The Octopus Resource Centre, have several modules of user interaction, one of them is made by the menu options, may be one of most used. Its building and working is based in Javascript language. Using the table of contents as an example (Figure 2), we can see that al the items are kept in data structures, present in memory of the local computer. When the user chooses a new menu option, that is connect with another sub-menu, an internal event is generated and managed by the Javascript programming, which obtains from the data structure the list of item from the selected menu and generates a graphic box with the displayed options.

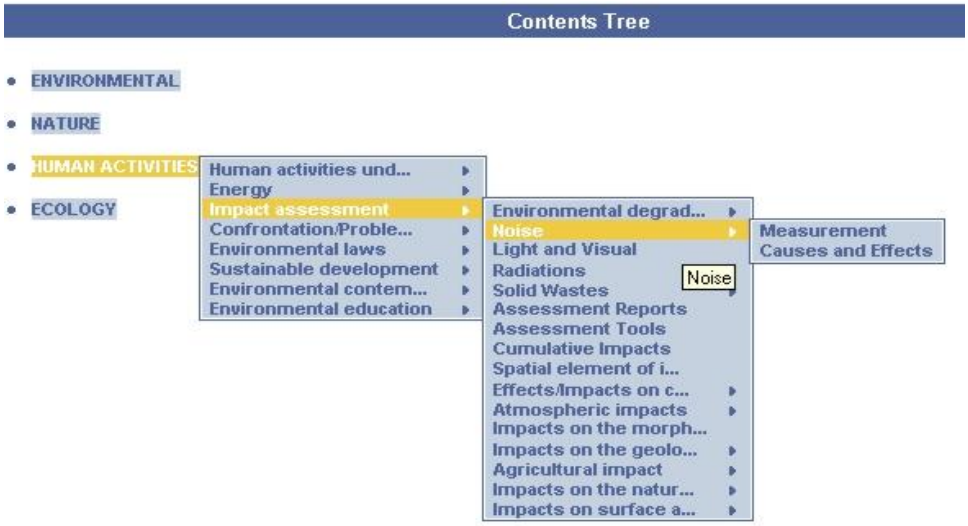

Figure 2– Table of Contents menu

The Java programming language, have as main characteristic, the fact that being interpreted, emulating a virtual machine and basing in a model oriented by objects. The first characteristic allows its use in different operative systems and hardware platforms. This way of working is very good for Internet using because, in this network, coexists many computing systems, so the Java programs can work in everyone. Using these characteristics, it was developed for Octopus, what are called applets, which are programs that can be executed by Internet browsers. These applets are embedded in the HTML pages and when the browsers charge them, have the behaviour of an ordinary software program. The applet technology was used in Octopus Resource Centre, mainly in the document editor and in the Learning sequences editor.

The development of these editors supposed a great challenge because have added difficulties, comparing with off-line versions, caused by applets action restrictions, in assuring security to the users that charge them in browser. The main restrictions that we had to overlap were related to reading files from the local computer disk (image files, PDF, DOC, among others), read and write in clipboard (to assure the copy/paste actions) and the access to restricted properties from the system. To solve these problems the technique of signed applets was used, which allow to add a digital signature, identifying the author and allowing the users to accept or not, the added privileges that the applet provides. Each one of the functionalities the editors have, like the table management, must have an associated description of the HTML elements. This characteristic, oblige that all the graphic elements should be mapped in their own HTML elements. This operation is based in a parser (a syntax and lexical analyser) and in a loader, which allows presenting the structure of the HTML document to be saved. As example, we can see the Figure 3, that describes in document editor, the visual representation of a table and Figure 4, describing the HTML codification of same table, after its parsing.

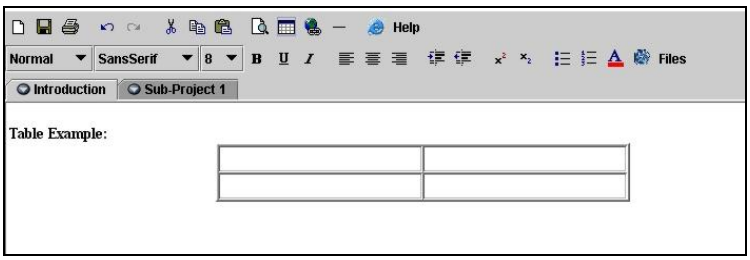

Figure 3 – Visual Representation of a table

```
<html>
<head>
<title>Table Test</title>
</head>
<body>
<table border="1" width="100%">
  <tr>
    <td width="50%"> </td>
    <td width="50%"> </td>
  </tr>
  <tr>
    <td width="50%"> </td>
    <td width="50%"> </td>
  </tr>
</table>
</body>
</html>
```

Figure 4 – HTML representation of a table

In the previous versions of the document and learning sequences editor, we had some difficulties in saving the learning sequences, which were built with some sub-projects and when were created in computers inside of institutional networks, controlled by routers and proxies.

We concluded that the process to save the MySql database data, didn't work well because it used a port of the database management system that was closed in many networks, by security reasons. To solve this problem we had to use another server, based in technology XML/RPC [16], that have the function of receiving http requests from browsers, sent by applets or Javascript code.

These requests have the information about the tasks that should be done inside the server, like activate PHP scripts, to access and write in the database.

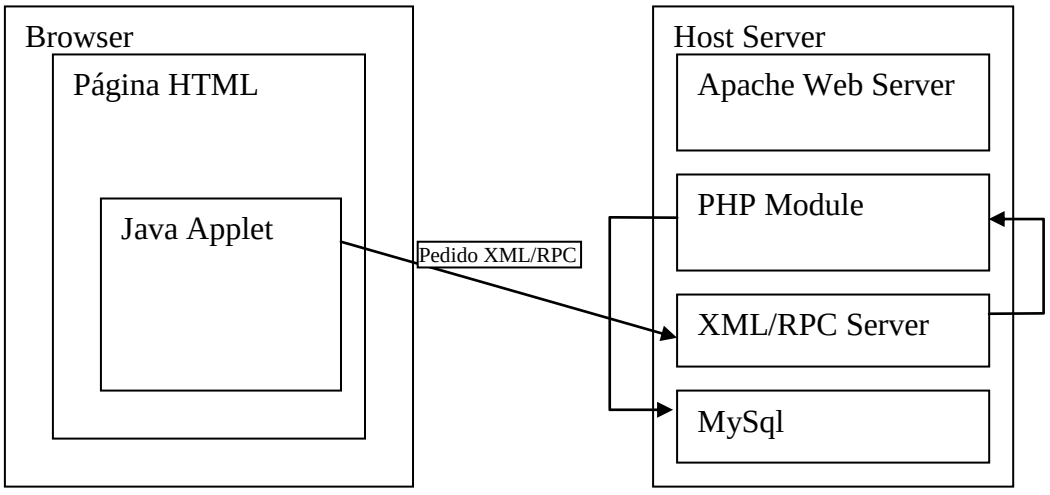

Figure 5 – System architecture with the use of applets and XML/RPC

We can see in the previous picture, that in case of saving contents inside database from the document editor (applet), it don't do it directly, but instead send the request and contents to the XML/RPC server, which calls the programming blocs of PHP, that do the rest of the work.

IV. THE OCTOPUS "STAGE"

The Octopus Resource Centre have available for its users, a great diversity of options and development tools, some of them we talked about. It's not the aim of this work to do a detailed description of all of them, but only to bring to light some of the most innovative issues, that came form the development of the project, giving the relevance to some aspects and technological solutions we used. It is import to make a general description of the information organization that is available at the moment, what surely gives us an idea of the Resource Centre Potentialities, of the complexity that we had to manage and from the difficulties of finding, in a short period of time a stabilization of the platform, mainly because new and more demands came to us to implement.

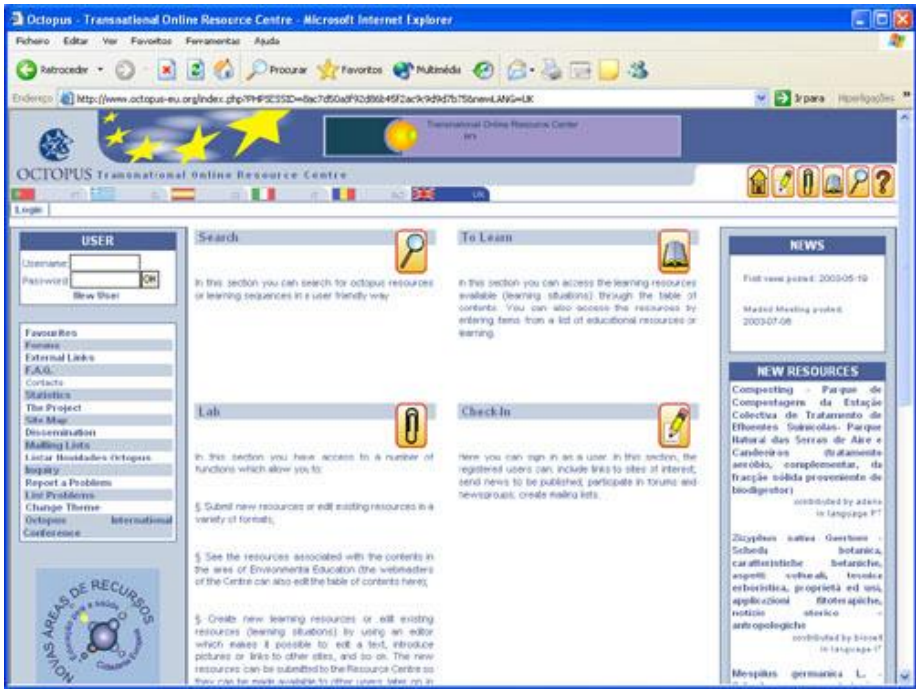

Figure 6 – Octopus Resource Centre Homepage

Reception

The name itself indicates the function. Its character is related with the task of receiving the users, to a great number of functionalities of the Resource Centre, namely the access to registration and to the available communication tools. The registration in mailing lists, in forums, news and new existing resources, are here possible, because all these tools are in this "room", so each user can integrate in the Octopus community, that all together can build.

Search

Several forms of search were defined, so each user, concerning to his needs and of the search specifications, always could find the needed information. This way four search forms were defined. The so-called *search*, the simplest, is just an introduction of a sentence in a field that verifies its occurrence. The *Advanced Search* uses more than a description sentence can be used in several fields, as author or organization. *Metasearch*, is the most complex, because the user can build his search, choosing the fields where he wants the search will apply. This mechanism allows the user to build a search using a composition between fields and logical operators. The fourth one is the *Directory Search*, which uses the structure of the Octopus Table of Contents to select its elements where the search will act.

Lab

In this room, are accessible the tools that allow the management of the Octopus resources. To these resources only the registered users, integrated in a high level profile, with enough permissions that allow sending information to Octopus, have access to them. With the existing tools they can change the table of contents, or build learning resources.

To Learn

This is the "room" mainly related with the access of the Octopus Resources. Here we can find lists of the Table of Contents, the base of all the registration and submitted resources, the list of the existing resources submitted to Octopus and the list of all the learning resources.

Help

We talk here especially of help, because it have an innovative aspect. It was developed in a dynamic way, fact not usual in Internet sites, so it can grow in parallel as Octopus. To build it with these characteristics, we gave it a context character, so each option can have its help file. This means that each new added option needs

the correspondent help that must to be built. Because of the separate working by languages, also with different helps, we kept the philosophy of showing the English information help while there no other exists in the native language.

V. "BEHIND OF CURTAINS"

Menu Manager

The possibility of the users could manage the options of accessing to information, was based in a menu managing system, that allows to define each option and its hierarchic relation with the others. After accessing the list of existing menus, its selection allows to define some properties.

Figure 7 – Octopus Menu Editor

It's in this screen (Figure 7) where the management of the information link is done, after selecting the menu.

This selection gives access to a new screen (Figure 8) where the help file is charged.

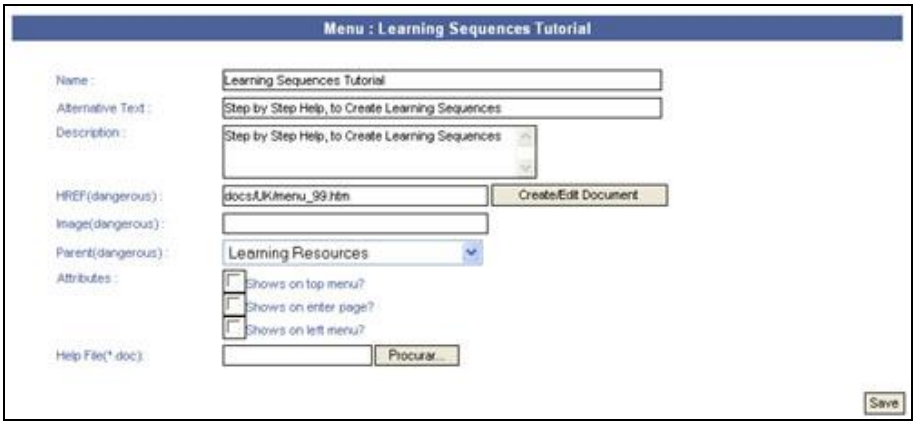

Figure 8 – Menu Learning Sequences Tutorial

Help Creation

The charging of help, of the accessed menu, is made by the upload of a DOC file; built in Microsoft Word format. We adopted this option of work, because these files have the means to generate and use internal links, which has been thought of great importance, especially in long file helps. In order of work with this kind of files, we develop ways of making conversions of DOC files to HTML format, in an automatic manner, so the easiness of work in charging, reading and using the help could be assured.

To do the conversion operation it was needed to setup in the server, a library of functions, which have the role of doing the parsing of the information structure, present in the Microsoft Word documents (formats 2000, 97, 95 and 6) and based in it to load the correspondent HTML format representation. This library is called WV [17] and is written in the programming C language, also based in a public license system. After the code analysis we realized that some functionalities didn't work, as we wanted, like the internal and external links. This implied a new challenge because it was needed to analyze all the structure of DOC files, to allow making the necessary changes of the programming code and to map the HTML code, needed to a correct conversion.

Document Creation Editor

The place identified as HREF in the Figure 9, is the one where is put de information access definition that the user want to connect. The information that can be put here, is so flexible, that its possible to have, a programming modulo reference, which was made separately; it can be a direct connection to an outside Internet site or it can be the connection to text document, a user wants to generate in that moment. To do it, he needs to press the button Create/Edit document, which allows the access to an editor, specially developed, being the preferential tool to introduce the information. This editor is used on-line, writing directly in Octopus Server.

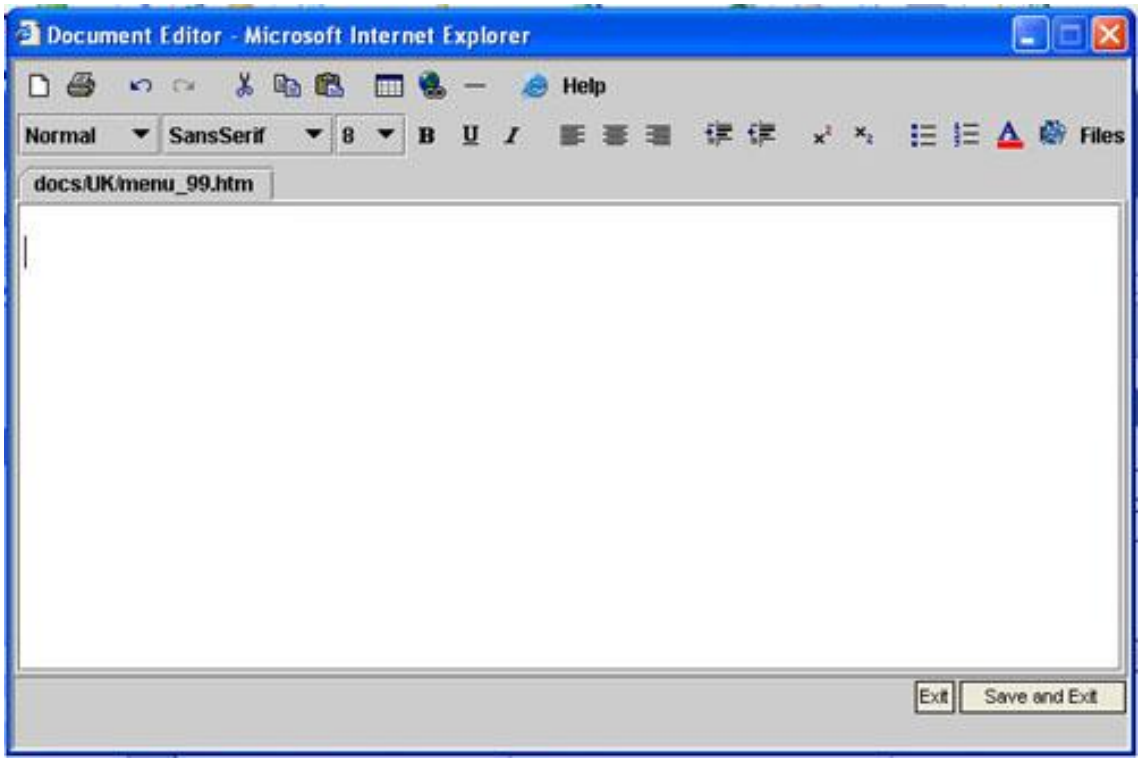

Figure 9 – Octopus document online editor

As we can see, the existing tools and the looking of the editor are quite similar to the ones of Microsoft Word. This option was made because of the knowledge of this program for many users. In the development of this editor, which was totally build for this project, based in Java Technology, we always had in mind that many users are familiarized with work routines that come from the using of Microsoft Word, so the same kind of functionalities were developed to incorporate them in it. We just want to say, in a very clear way, that all of the edition work is made on-line, so in spite of many tools being present, is very difficult they have the sophistication of working as if they were off-line, at least with the actual kind of slow internet accesses.

It wouldn't be useful and boring to describe all the tools inside of the editor, because most users know most of them. It is important the reference to the use and the interaction with Octopus resources, as with off-line files, of a user's personal computer. In top right side of the editor, it's present the Octopus logo, which represents the access to Octopus server resources, after the opening of a new window:

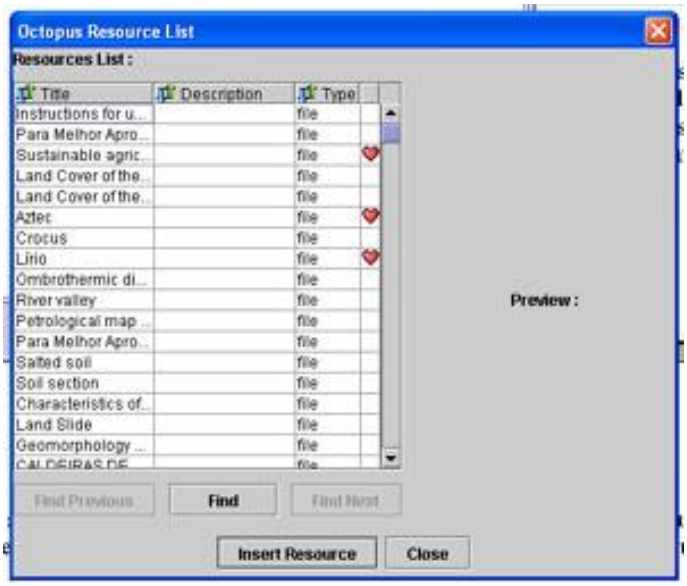

Figure 10 – Octopus resource list

The selection of one of them, and the pressing of Insert Resource button, has the consequence of putting the resource in window editor (Figure 10). If we selected an image resource, it will be showed directly in the editor window, if it's a text, it will be included as a link. To allow the same kind of working with the existing information of a user's personal computer, also was developed a personal resource zone, which is only available to the users with high-level permissions, because this kind of personal resources are only destined for building documents. Its access is made pressing the editor's Files button, which opens a new window with this list (Figure 11). To these kind of personal resources, although they can have a common use, its not applied the registration and classification form.

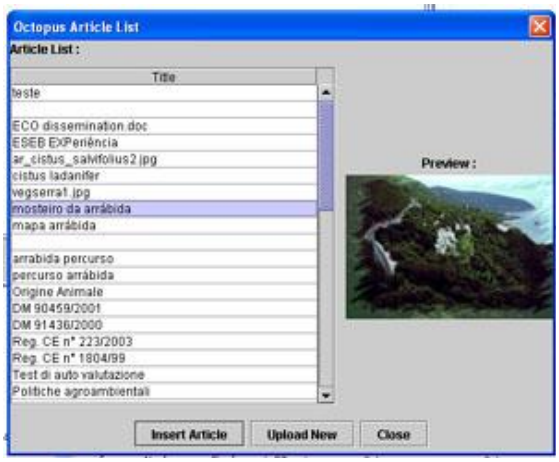

Figure 11 – Octopus article List

Octopus Internationalization

Previously we already explained how Octopus Resource Centre, works with the several languages, and the needs for a correct use and some standards to a correct articulation between them.

This way of working didn't solve all the problems concerned with the uses of different languages. The internationalization process had to be assumed in a more global way, because we must assure that all parts where there was some information, it could appear in the different languages. The global approach of this problem allowed us to identify three distinct areas where we must act. We can call them the **Information** area, concerning to all options and menus; the **contents** area, concerning the documents and the **code**, related with all the messages embedded in the programming code of the site.

The translation of the information area was possible by the use of menu editor system, as we previously wrote. Concerning to the contents, we assumed it in two different perspectives. In case of the written documents, the translation was made possible by the using of the documents editor, which is the common tool to the high level permission user do the

work. Concerning to the high diversity of resources submitted to the Resource Centre, it was defined that each author has the power of decision[1] in which language the resource would be. Knowing that this would cause the existence of many resources just in native language, our concern was to have a common identification base. This was made by having two obligatory registration fields, one to mark the language which the resource is destined, the other to write always in English, the correspondent name of the one in native language.

It's procedure allowed to have a common base to broadcast the information[2] about the submitted resources, having also consequences in search, because commonly the language is an important search factor.

In case of programming code, the translation process was more complex, because the programmer couldn't reply each code section in a different language. It would be an enormous work e almost impossible to assure the information integrity. So, the found solution was to make the identification and marking of all the sentences containing warnings and messages embedded in code.

After this work is finished, it was developed a translation program that after identity all this marks, creates a list with these. This program was sent to the partners with the list of sentences in English, so they could make the translation.

The translated lists, in each language, were imported to a correspondence file that makes the translation to the selected language, of the marked places of code, when it's executed.

To have this functionality working was needed to activate the Gettext interface, an extension of PHP, specially related to the internationalization tasks of code. This interface accesses to a binary file in .MO format, which contains inside of it, all the strings to translate, indexed by a key.

To access the translation of strings in the different languages in the .MO file is needed to send to the translation function, the arguments, *language* and the *string* that we want to be translated.

The program that allows the partners to do the translation tasks was built in C++. The translation catalogs of each partner, were saved in a .PO file, and after all the receptions made, were compiled in the format .MO.

We must tell that this process had some problems, related with the Greek translations, which have characteristics very special in the definition of its characters, because they have a different codification from the other western languages. To solve this problem some changes in the code were made to accommodate the exceptions related with translation of Greek language.

Learning Sequences Editor

In the Octopus Resource Centre, there are simple resources, which have been submitted without concerns about the use it will have and other kind of resources, which are destined to promote learning. In the project team we called them Learning Sequences. We can consider them as complex resources, because they were build having defined aims to achieve and also because they obey to some rules of construction and have mixed several elements.

To give an answer to these needs the Documents Editor was expanded, so it could have new tools and functionalities.

Each of these Learning sequences, have to work as a little project, where some tasks are asked to the users do, and they need to explore a list of selected resources (from Octopous, from internet or other), based on a specific problem or content.

This "big project" is sub-divided in different approach perspectives or different tasks, which are accessed by using a set of folders.

To have these possibilities we had to make available, to the sequences creators of an edition tool that allows the creation process and the publication of created sequences in the Octopus Resource Centre.

The Learning Sequence Editor is the result of the development related with this needs. After putting some questions to the user with high-level permissions, it builds a model of sequence that is filled up by the user.

Each existing folder in the sequence have the possibility of changing it's properties, as the identification name and the sequence order where it's placed. Also the user can delete and insert folders, being free not to answer to the initial questions that were the base for the creation of the initial model of Learning Sequence.

The editor is enough flexible to allow that creation process is extended by several steps. It could be useful, because this creation can involve many time for some research and building work, so the final user, don't have access to the learning sequence until his creator don't submit and make its registration.

VI. CONCLUSION

The Octopus Resource Centre is an Internet site, that is the result of a good balance between the initial idea, the need of identifying the requirements of the project, and the identification of needed technologies to have results.

In spite of many conditionings, in the development process, the final result is very promising, because it allows different perspectives of work and uses and also a big increase of information in terms of submitted information and using.

After two years of intense work, the technical limitations are surely the least important, because the existing tools and the great flexibility of Octopus site

[1] We are mainly referring to text documents, because other kind like photos, it have no meaning to talk in destination language, or in building language.

[2] The broadcast of this information is made in home page, in the section New Resources, where all the resources titles are in english (if the registration rules are followed), no matter what language is selected.

uses are the result of a great concern with the final user. Not only the information consumer, but also and mainly, with authors of new information to include in Octopus. For these and for their needs some tools were specially developed and sometimes very complex problems emerged. The problems became challenges that obliged us to make difficult and risky options, but gave us also the opportunity to open new routes.

The Octopus Resource Centre, became in it development time, a great and complex amount of programming, just comparable with some institutional sites as banks, where a a few thousands of programming code lines were used. It was possible with relatively few means and many creativity to have an Internet site very powerful, very complex, but mainly very flexible, that we can say without many presumption, a mark in the development of on-line resource centres.

REFERENCES

[1] CARIOCA, V. e **PASSARINHO, A.** (2002). *Project OCTOPUS-On Line educational resource centre, a metaphor for a hypermedia environment*. Paper presented at III CONFERENCIA Internacional Virtual Educa 2002,Valencia (12-14 Junio). In Actas de la conferencia (CDROM para windows - ISBN 84-930206-3-x) - http://www.aefvi.org

[2] CARIOCA V., GIURGIU, M., PASSARINHO, A., GAIBINO, R. e CHOURIÇO, J. (2003). *OCTOPUS Project: from alfa to beta version*. Paper presented in I ENCONTRO IBERICO DE TIC, Escola Superior de Educação de Beja, Janeiro. In Actas do Encontro, CARIOCA,V e ALMEIDA, C. (Coords.) EVOLUTIC 2003 (CDROM- ISBN 972-95970-1-4)

[3] MEDINA A.; DOMINGUEZ, C.; (2004) Center of resources octopus in environment education Pedagogic reference: sequence of learning, http://www.octopus-eu.org/enter.php?page=main.php?id=25

[4] Linux at http://www.linux.org/

[**5**] Unix at http://www.unix.org/

[**6**] MSWord at http://office.microsoft.com/

[**7**] HTML at http://www.w3.org/

[**8**] Apache at http://www.apache.org/

[**9**] Netcraft at http://news.netcraft.com/

[**10**] Mysql at http://www.mysql.com/

[**11**] PHP at http://www.php.net

[**12**] Javascript at http://devedge.netscape.com/central/javascript/

[**13**]Java at http://java.sun.com

[**14**] Cascading Style Sheets(CSS) at http://www.w3.org

[**15**]Netscape at http://www.netscape.com

[**16**] XMLRPC at http://www.xmlrpc.com/

[**17**] WVware at http://wvware.sourceforge.net/