

Chamilo 2.0: A Second Generation Open Source E-learning and Collaboration Platform

[doi:10.3991/ijac.v3i3.1364](https://doi.org/10.3991/ijac.v3i3.1364)

Jean-Marie Maes

University College Ghent, Ghent, Belgium

Abstract—Dokeos used to be one of the leading open source learning platforms. It is well known for the wide range of functions it offers and especially for its ease of use. In many ways it is however a typical first generation learning management system (LMS) consisting of a set of loosely integrated tools. The original design of this LMS has a number of serious drawbacks and limitations. Because of this it was decided to opt for a completely new system that would use state of the art development techniques. The new version is built on a repository of reusable simple and complex content objects, not only documents. The web LCMS is only one of the possible applications that run on that repository. One of the other applications that can make use of the same set of objects is the portfolio. Functions that used to be modules within a course can also simply be used as applications, e.g. forum, assessment, wiki.

All functions such as creating, changing, editing and deleting an object are basically the same platform wide. As a result all modules and applications interact in a far more consistent and less bug prone way. This also greatly simplifies the user interface. This interface that is common to all objects in all tools and modules, makes a number of publication features available that were not available in all tools before, such as categorization, selection of target groups or users, visibility and period of publication. This is certainly adding to Chamilo's ambition to be a user friendly, flexible, fast and consistent platform.

The adaptability of the new system with integrated portal functions and a framework for web services makes it a suitable instrument not only for universities and schools, but also for the business world. In this way Chamilo offers a platform that is ready for future developments in a globalized world where life long learning and collaboration will be central.

Index Terms—Chamilo, life long learning, LCMS, open source, repository

I. INTRODUCTION: FROM DOKEOS TO CHAMILO

Dokeos used to be a widely used open source learning management system (LMS), especially in Europe and Latin America. It is known for its wide range of functions it offers and especially for its ease of use. In many ways it is however a typical first generation learning management system (LMS) consisting of a set of loosely integrated tools. As is the case with many other LMS's, its original design has a number of serious drawbacks and limitations. The design is course centered: data is confined to courses, there is no central content management system that stores all learning objects including documents, various types of

assessment, and all possible types of simple and complex learning objects. An essential limitation is the limited re-usability of learning objects. A document e.g. that is uploaded into a course, has to be uploaded a second time if it is to be used in another course. For these reasons proposals to write a completely new version 2 were put forwards a few years ago. In the mean time there was a growing dissatisfaction within the users and developers communities with the lack of structure and with the choice of Dokeos company for commercial releases with only a limited free version. Dokeos was clearly evolving towards a mainly commercial application that was completely controlled by Dokeos company. Many users and developers had initially opted for Dokeos as an open source environment and they were disappointed about this evolution. This finally led to a split in January 2010: the large majority of developers choose to go with the new open source project Chamilo.

Chamilo is structured as a non profit association according to Belgian law. The association brings together all stakeholders involved: companies as well as universities and other educational institutes, either as developing partners or as users.

In many ways Chamilo is the logical successor to Dokeos, certainly as a fully featured open source platform. This was clear from the first release Chamilo 1.8.6.2, which was the immediate successor to Dokeos 1.8.6.1. This has given Dokeos users the possibility to move smoothly to Chamilo. Another release has followed in May 2010, Chamilo 1.8.7. Next on the agenda is the release of the completely new Chamilo 2.0 with a Beta version by the end of June and an official release some time later.

II. CHAMILO 2.0 BUILT FOR THE FUTURE

It was no easy option for an existing software community to go for a completely new development from scratch. It has been our aim to build a web based learning and collaboration platform that would be easy to use in a variety of contexts, easy to integrate, to connect to and to extend. The re-usability of resources is the key feature of Chamilo 2. This is also the reason why the course metaphor is no longer central to the platform. In fact this metaphor is outdated and too restrictive even in mainstream educational institutes such as universities. There is more to education than courses, there also is a need to accommodate working groups, projects, communities, etc. There is a clear need for educational resources in an electronic format to be reusable in the widest possible sense: not only within a platform, but also across platforms. Furthermore we wanted our new platform to be as generic as possible, making it

into a flexible tool that could easily be adapted to different environments, both in educational institutes of all kinds and in companies. The lack of generical design in present day electronic learning platforms can be seen in the totally different variety of platforms used in the educational field and in companies. Blackboard e.g. has a leading position in education, but is barely used in companies.

A. Chamilo 2.0 architecture

When designing the new architecture we had a few keywords in mind that had to lead us in our choices. We wanted an architecture that fully allowed for

- flexibility: easily adaptable to different contexts;
- modularity: it should be possible to activate only what you need in a particular environment;
- extensibility: it should be easy to build additional applications;
- from a more technical point of view we wanted the system to be
 - object-oriented, referring to the way the code is built up;
 - generical: code should as a rule not be specific, but allow for easy extensibility.

That is how we have come to a layered model (Fig. 1). This layered approach consists of four supporting layers around the object repository. The object repository is the basis of the system and it holds all available data. On one side of this layer sits the database abstraction layer, which enables us to use a wide range of database systems underlying the repository. In the same way file system abstraction is used to be able to work with various file systems. The repository data manager is the link to the access layer, i.e. the various applications running on top of the system.

This access layer offers roles and rights management to the different applications which are considered to be locations within rights managements. Locations are stored per application in a nested tree. Locations in combination with a certain 'right' are linked to (one or more) groups, users and templates. This allows the system to check whether a user has the proper access rights to manipulate an object in the requested way.

On top of these layers various applications can be implemented, like a course management system or LMS, a portfolio system and other systems or applications.

1) The object repository

The kernel software is mainly formed by the object repository. This is the basis of the platform. The repository data manager links up with the various applications. The applications can be switched on or off depending on context and needs. On a technical level the kernel software uses abstraction layers to work with a number of database and file systems.

The object repository is a good example of the generical nature of the system. Contrary to other learning environments where the repository is mainly limited to documents, the Chamilo repository accommodates for all possible objects. These objects are no longer strictly 'learning' objects, but can be content objects of various types. This clearly shows the generical nature of the platform, making it suitable for all kinds of contexts, be it educational or business.

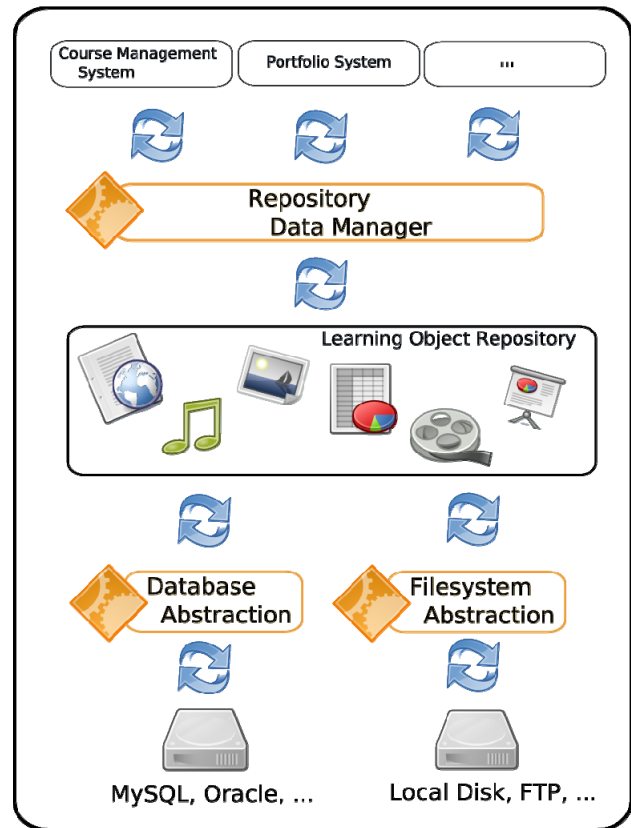


Figure 1. Chamilo architecture

There are basically two types of objects: simple objects that cannot be reduced further to any constituent part and complex objects. Complex objects consist of any number of other objects, both simple and complex. All of these objects can be shared and (re)used throughout the system. Objects remain in the repository. They are not copied to any application running on top of the repository. So contrary to the first generation LMS's all objects conform to the motto *store once, use many*. Microsoft Sharepoint a.o. is an example of how difficult it often proves to put that nice principle into practice.

Changes to a specific content object will be immediately and consistently available in all those applications, e.g. changes to a document only have to be made once and the changed document will be immediately available in all courses where it is published, but e.g. also in the personal portfolio or any other application if published there. Objects come with a versioning system, adding even more flexibility to objects and their publications.

2) The user object repository

The repository is user based, meaning that any content object is owned by the user that created it. Each user has a view of all the objects he or she owns. This view very much resembles the tree like view of a hard disk with which any user is familiar. The user repository can be considered to be something like a virtual disk. The user can export his personal repository in one step and e.g. import it into another Chamilo platform. This is the first step towards making full portability possible at a later stage. As after some time there can be very many objects in a user's repository, it is possible to view the repository by object type (this is not a closed set of object types, types are extensible as most elements are within the sys-

tem). Furthermore a user can define a set of personal views, e.g. any objects related to assessments or to surveys.

There is much more of course. All content objects e.g. can be set searchable on the basis of their metadata even if they are not shared.

3) *Sharing content objects from the repository*

As developing electronic content material is still very much a time consuming task, being able to search for similar content objects is very important. In that way you can come into contact with other users trying to develop similar objects. The search portal now already offers basic search functionalities all over the platform and even across Chamilo and other e-learning platforms and repository systems (Fedora Commons Repository is already supported).

Objects in the repository can standard be set to searchable on the basis of metadata. They can also be set to

- readable: allowing others to view the content, i.e. the object itself,
- usable / shareable: allowing others to use, i.e. publish the object and
- reusable: this allows the (new) user to reuse the object and make changes to it, turning it into a new object of its own.

The owner of an object is able to select users and groups of users to link to the sharing features of an object. In this way any object becomes searchable for anyone in the system (or for that matter for anyone in the world if this setting is allowed on the administrator level). Next to that it will be possible to mark an object as readable for any number of users or groups of users and usable or reusable for other users or groups of users.

Such a potent search function can be expected to facilitate closer cooperation of instructors within the institute, but it will also enable the user to work together with anyone in his or her field in many systems worldwide. In this way new communities of knowledge can be expected to grow fairly spontaneously.

4) *Applications and publications*

a) *Core and non core applications*

Applications run on top of the repository. We can distinguish between core and non-core or optional applications within Chamilo. Core applications (we also call them libraries) are those that are relevant to the entirety of the system. They can be used for all types of web applications. They use the Rapid Application Development (RAD) model. Examples of the core applications are user and group management, rights management, administration, tracking, reporting and many small functionalities that can be used in any PHP application, such as home page, menu and web services (e.g. for authentication, with a.o. support for CAS, LDAP and Shibboleth). They are built as generical frameworks that can easily be extended.

The core is in fact a RAD framework for PHP. This framework is relevant to any non core application. Consequently core applications cannot be switched off, non core applications can be switched on or off according to the needs of any particular installation. These non core applications are mainly the applications the regular end user is going to see and use. To this user they form the essence of the platform. They include the LMS, or simply put the

‘courses’, but also applications such as search portal, portfolio, reservation tool, profiler, personal messenger, personal calendar and web conferencing (Dimdim integration). Furthermore a number of tools that used to be restricted to the courses module can now be used as autonomous, platform wide, applications: forum, gradebook, wiki, assessment (including competency based assessment) and survey.

b) *Publications*

The repository data manager manages the ways in which objects are used, i.e. published, in the various (non core) applications. On a general level an application can be considered to be a user interface in which content objects from the repository can be published. In the future it will be possible to further introduce new types of objects and new applications or views on the repository without much hassle, as all basic tools are already available and only things that are really specific to the new object type or application need to be developed. In fact there is already a framework available to build new applications easily.

As a result all course modules and applications interact in a far more consistent and less bug prone way as they share the features for a number of actions that are dealt with by the repository data manager. This consistency is a major advantage to the end user. In fact a user will see the same basic set of pages for all content objects in all tools and modules. They will enable the user to perform a number of actions such as:

- creating a (content) object
- editing an existing object
- deleting a object
- publishing an existing object
- searching for a object.

The same goes for categories of objects.

This has enabled us e.g. to easily develop a search function, making it possible to search for objects within a particular Chamilo platform or even across various Chamilo platforms. This search is based on metadata.

A publication can have a number of elements linked to it, depending on the application where it is published. When e.g. publishing one or more documents to a course, you can select a number of publication details (fig. 2), such as the selected users or groups of users, the publication period, whether the object is hidden or viewable, whether an e-mail should be sent notifying other users of the publication and whether the objects should be put on the homepage. All the characteristics do NOT belong to the object itself, but to the specific publication in a certain location within an application. When an object is published it is usually (again depending on the application and the type of object) possible to comment on it, or to create feedback. These recurring publication details have helped making the user interface more uniform and they have made a number of publication features available that were not available in all tools in previous versions, such as categorization, selection of target groups or users and period of publication. This has certainly added to Chamilo's ambition to be a user friendly, flexible, fast and consistent platform.

The same object could be published in various courses or course modules, it could be part of a complex object

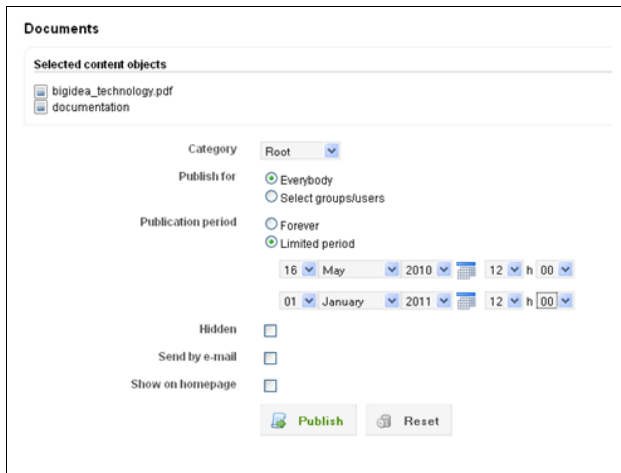


Figure 2. publication details

such as a portfolio, or could be attached to one or other feedback page. The object in all these examples remains the same. It is stored only once in the repository, but it is published a number of times in various applications. The publication details can be different for each publication and they can be edited individually for each context.

B. A growing range of applications

As was mentioned before Chamilo 2 has a lot more to offer than a classic LMS. The fact that quite a number of applications can now be used platform wide without the need for a ‘course’ or ‘community’ to be created first, adds a lot to the flexibility of use. As it is relatively easy to add new applications to the system, quite a few of them are now in the pipeline and a number of existing ones will

be much improved. Work in progress includes improving the assessment application. Competence based assessment and peer assessment are being built out. Furthermore quite a number of new question types are being added to the assessment and survey applications, including powerful types such as the matrix question type. New applications include an internship planner, a student monitoring system and various small applications e.g. with regard to electronic library use.

C. Chamilo 2 portal function

Chamilo 2 is much more than a Learning Content Management System (LCMS). The kernel software makes it relatively easy to build new applications. It is also easy to connect to outside applications through the framework of web services that is built in. Therefore it was only logical for the home page of Chamilo to have portal functionalities. The user can decide on which elements get a prominent place on this portal page (Fig. 3). Those among you that are familiar with i-Google will understand how it works. In fact at the time of development i-Google did not exist yet. It was the BBC-website that inspired us for the Chamilo portal. Basically the user can put a selection of various Chamilo items on this home page such as the course list, new announcements, the personal calendar, etc., but also RSS-feeds and the list of possibilities is certainly going to grow fast after the release.

The administrator has of course more options. He can define the menu tabs and all elements of a menu and their order. In the LMS application the various course types have a wide range of settings, including several menu settings. A course administrator can e.g. enable a collapsible left or right menu with either text elements, icons or a combination of both.

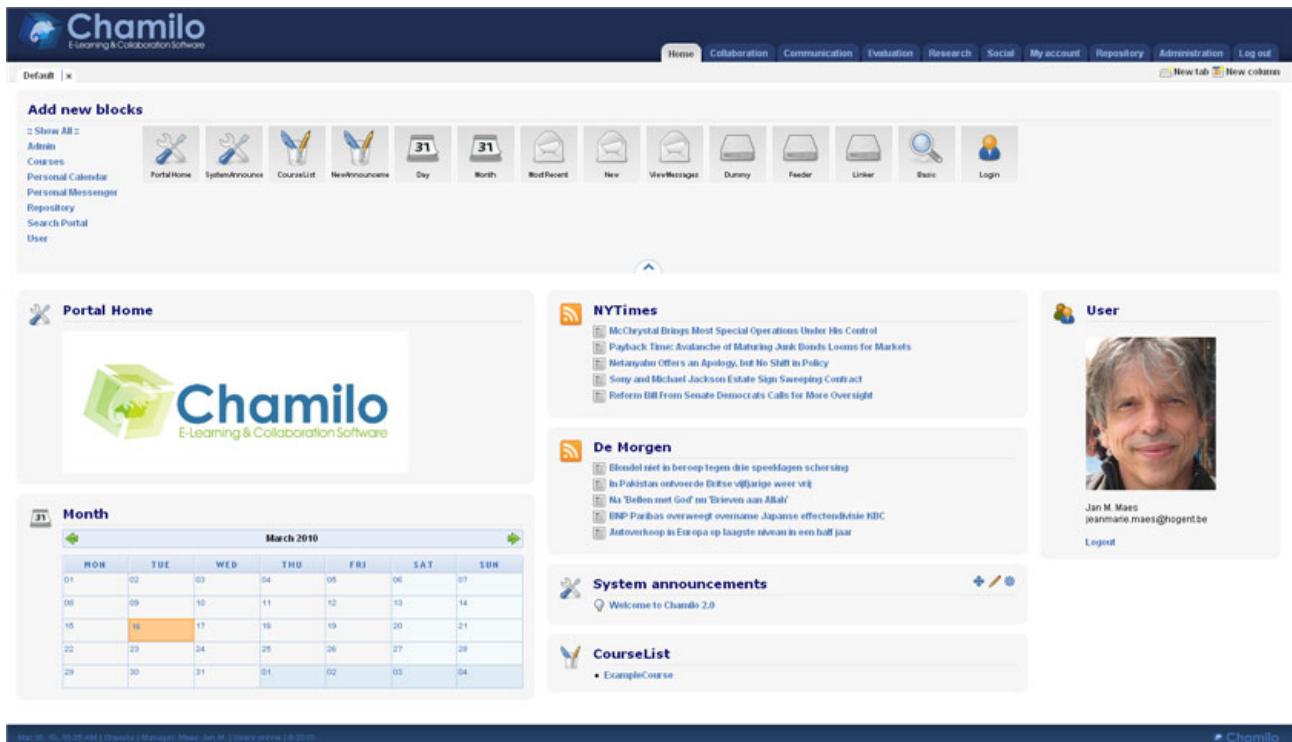


Figure 3. Building your own Chamilo portal

D. Chamilo 2 and beyond

1) Chamilo Association

The evolution towards closed software of Dokeos Company and the lack of any decision making structure involving the development community was the basis of the split of Chamilo from Dokeos. To ensure our choice for open software development, the large majority of the developers and users communities have decided to start the Chamilo Association. Chamilo was officially launched in January 2010. It has established itself very quickly as the successor to Dokeos. Activity has clearly increased as can be seen in Fig. 4 where Chamilo (red), Moodle (green) and Dokeos (blue) are compared. (source: <https://www.ohloh.net/>):

It is particularly nice to note that Chamilo is now performing slightly better than open source LMS market leader Moodle, both as far as activity (number of commits) and number of contributors is concerned.

The Association is a non profit organisation according to Belgian law. Chamilo Association is dedicated to organising and coordinating research & development, promotional and recreational activities for the development and the promotion of open source e-learning and collaboration software, in particular of the Chamilo software suite. It is the owner of the Chamilo trademark and its General Assembly and Board of Directors are the decision making bodies for the whole Association, developers, users and interested companies. All of these are strict guarantees for the open source nature of the project.

There are basically two (legal) types of members with the association: effective members and admitted members. Only effective members have voting rights at the General Assembly. Admitted members can be invited to attend the General Assembly, but they then only have an advisory role. Effective members can participate in the various working groups. Admitted members can be invited to participate in the working groups, but they only play an advisory part. Local users groups are formed by Chamilo users grouped by the context of a region, country or language. Local user groups are very important for the spreading of the Chamilo software in the world, as they form the direct connection with the Chamilo end users. Common interest groups could e.g. be an accessibility group or a Chamilo teachers group. Both local user and common interest groups will be important for the growth of the Chamilo Association.

2) The future

a) Short term

The final release of Chamilo 2.0 is planned for the autumn of 2010. In June an Alpha and Beta version have already been released. It is planned to be released under the GPLv3 licence, more specifically the GNU Affero General Public License (AGPL).

Afterwards work will continue on a number of features that have not yet been fully developed in version 2.0. These are a few important short term targets:

- work on metadata;
- improved global search function;
- (further) implementation of e-learning standards such as SCORM, IMS, IEEE, QTI,;
- integration with Google docs.

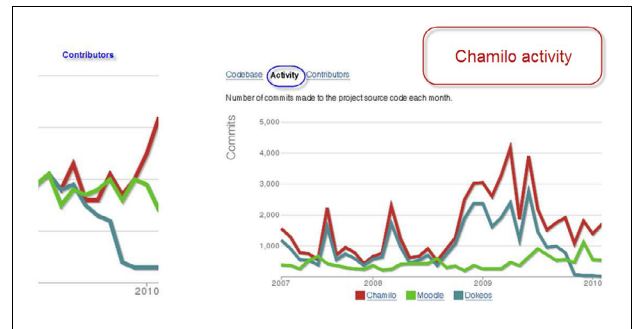


Figure 4. Chamilo activity & number of contributors

In version 2 a user can add metadata manually, but it is our aim to provide ways to automatically generate ‘intelligent’ metadata. Furthermore a number of ways to provide ‘social’ tags will be added. The work on metadata (based on IEEE LOM) will include:

- automatic creation of metadata using state of the arts tools;
- use of personal metadata profiles;
- use of existing information from course description (e.g. at University College Ghent: extensive XML-files).

Integration with Google docs, i.e. the possibility to directly publish Google documents to the Chamilo platform will greatly improve the usability of Chamilo as a platform for collaboration.

The global search function will be further improved with special attention to connecting to other free repositories.

We will also further adopt existing and evolving e-learning standards as far as these standards still make sense, especially e-portfolio standards.

b) Medium term

Chamilo 2 in itself is a completely new learning and collaboration platform. Its structure offers wide avenues of possible development in the future. It is difficult to predict what additional applications will be developed, esp. by third parties. In general we can safely foresee a personal portable Chamilo version in the medium term. Chamilo is built on a repository system and has all necessary tools to function as a web content management system (CMS). As a result it will be possible to use the system not only as a learning and collaboration platform but also as a website publishing tool. The last element will also be a key feature of Chamilo portable. This is a very logical next step to the present situation where a user can already export his own object repository. With Chamilo portable he will be able to publish these data to his personal website. Take a user who has built several portfolio’s during his educational and / or working career. When he is no longer connected to a specific educational Chamilo platform, he will still be able to further develop and publish his portfolio’s to the web with the portable version of Chamilo.

Chamilo portable will:

- be particularly suited for life long learning;
- offer integration of or connection to social software tools;

- enable users to continue using their personal learning and collaborative environment (portfolio, repository CMS, publication to personal website) without a connection to an institutional Chamilo site;
- plug-in & play: when needed their personal environment could be plugged into an institutional Chamilo (or in the long term any other standard learning environment).

In this way Chamilo will offer a lifelong learning and collaboration system that is personal and can connect to formal learning environments and social software at any time.

Lifelong learning will be possible as a really continuous process, without any loss of information at any stage.

The system will be flexible enough to be usable from primary school to various forms of adult education and to various working environments. For the first time a flexible

personal learning environment that can connect to formal systems in education and industry will be available.

This will have a potentially enormous effect on education and on educational efficiency both in traditional educational environments and in industry and business, esp. from the learner's perspective. It will integrate learning into anyone's personal life as a integral part of that life.

AUTHOR

Jean-Marie Maes is with the University College Ghent, Belgium. He is a member of Chamilo Association (e-mail: jeanmarie.maes@hogent.be; website: <http://chamilo.org/>).

This article was modified from a presentation at the ICELW2010 Conference in New York, USA in June 2010. Submitted July 21st, 2010. Published as resubmitted by the authors July 29th, 2010.