

# Towards an online video platform for MOOCs

Pantelis Balaouras<sup>1</sup> · Michalis Gatzonis<sup>1</sup> ·  
Konstantinos Tsimpanis<sup>1</sup> · Hippokratis Apostolidis<sup>2</sup> ·  
Thrasylvoulos Tsiatsos<sup>2</sup> 

Received: 26 October 2016 / Revised: 12 April 2017 / Accepted: 5 June 2017 /  
Published online: 12 July 2017  
© Springer Science+Business Media, LLC 2017

**Abstract** This paper presents the OpenDelos platform implemented in the context of the project “Central Repository of Greek Open Courses”. OpenDelos is an online video editing and publishing platform for online courses, e.g., Massive Open Online Courses (MOOCs). OpenDelos is a unique open source platform that supports not only video on demand functionality but also live streaming functionality along with the related scheduling functionality. OpenDelos is suitable for Higher Education Institutions (HEIs) offering (a) academic roles structure, (b) unique data presentation of the (c) searching and browsing features, (d) hosting of open multimedia and educational content. The main goal of this paper is to present the technical aspects of OpenDelos platform. Thus, this paper describes OpenDelos’ architecture, modules, and implementation issues. Furthermore, the heuristic usability evaluation conducted by experts during the implementation of OpenDelos platform is presented. The heuristic evaluation showed that OpenDelos did not present major usability problems. Experts’ comments have been taken into account to improve the platform. The improved version is ready and it has been distributed to the end users in order to measure end users’ acceptance towards OpenDelos.

**Keywords** Video platform · Live streaming · Video editing · MOOCs

## 1 Introduction

A recent popular trend in e-learning is the deployment of MOOCs (Massive Online Open Courses) [11]. MOOCs are a continuation of the trend in innovation, experimentation and the use of technology initiated by distance and online learning, to provide learning opportunities for large numbers of learners [11, 41].

---

✉ Thrasylvoulos Tsiatsos  
tsiatsos@csd.auth.gr

<sup>1</sup> National and Kapodistrian University of Athens, Athens, Greece

<sup>2</sup> Department of Informatics, Aristotle University of Thessaloniki, GR-54124 Thessaloniki, Greece

Most of the Greek Higher Education Institutes (HEI) - universities and technological education institutes - have launched projects for developing open courses, which were funded by the Greek Ministry of Education in the context of the Operational Programme “Life Long Learning” [33], co-financed by Greece and the European Union. A typical open courses project at National and Kapodistrian University of Athens (UoA) is presented in [27]. After three years of development, more than 3.700 high quality Open Courses, with common specifications for 25 HEIs, are available to the general public via the National Open Courses Search (NOCS) portal developed by the Greek Universities Network (GUnet) [17, 18, 46].

Nowadays, the video constitutes a major type of educational resources. MOOCs are based on multimedia collections that include the video, audio and slides of the instructor(s). The developed Greek Open Courses are in line with this trend. Therefore, many of them (1100 of 3700) include educational videos, such as video lectures. More than 20.000 video files have been produced in the context of the Open Courses.

The need for a video platform capable of hosting, processing (online), managing and publishing this large number of videos is obvious. Therefore, GUnet has designed and developed the OpenDelos platform [32], a rich media and lecture platform for Education.

OpenDelos’ innovative aspects - an open source platform with online video editing, slides synchronisation, live streaming support, tagging content with open licencing, etc. - allowed the Greek HEI to develop and host a significant number of videos as Open Educational Resources (OER). Its open architecture allows for scalable deployment without financial or technical restrictions giving the chance to the Greek HEI to concentrate on the content development process.

The impact of OpenDelos platform in video based learning provided by Greek HEI is of great significance. Most of the Greek HEI, have installed the OpenDelos platform (21 installations) for hosting their video, more than 25.000 video files are hosted by OpenDelos platforms. A best practice example of exploiting OpenDelos is the OpenDelos installation of National and Kapodistrian University of Athens [47], which hosts more than 15.000 videos and provides live streaming from 55 classrooms.

This research work follows a holistic approach focusing on the design, implementation and evaluation of the OpenDelos video platform. More specifically, Section 2 “Video platforms: functionality and requirements” discusses the value of the video based learning that leads to the necessity of deploying video platforms, especially by educational institutions, introduces the video/lecture capture systems and video platforms, presents the main categories of the video platforms functionality, and, finally, elaborates of the functional requirements that have led to the decision of design and implementing the OpenDelos video platform. Section 3 “OpenDelos Design: Architecture and Implementation” presents the architecture of OpenDelos platform and implementation issues, that is, how the OpenDelos functionality has been developed explaining major technical issues and decisions. Afterwards, Section 4 “Heuristic Evaluation” presents the heuristic usability evaluation results of the OpenDelos platform in order to resolve any usability issue. The final Section 5 presents the concluding remarks and the vision of the research team for the next steps.

## 2 Video platforms: functionality and requirements

This section presents (a) the significance of video-based learning; (b) lecture capture and video platforms; (c) functionality of video platforms; and (d) functional requirements for the video platform.

## 2.1 Video-based learning

Exploiting video transmission technology for developing class sessions cannot be characterized as a recent trend. In opposite, the video-based learning has a long tradition [6, 37]. Therefore, assessing the impact of video on the learning effectiveness, teaching methods and tool design has also a long history in the research. Studies has shown, from the early beginning [9, 54], the value of video, especially of the interactive video, in the learning performance. Recently, the interest on how to produce, host, distribute and use of online learning videos is increasing, since online open courses are regarded as an important method to support education [16]. The research on video-based learning is reviewed by [16, 21, 53].

## 2.2 Lecture capture and video platforms

Minimizing the cost of producing and hosting online learning video is important for achieving sustainability of the educational services that exploit video. The need for decreasing the video production cost (camera crews, video processing) has led to widespread deployment of lecture capture, especially within higher education, by exploiting automated audio visual systems [5]. These systems are automatically transferring the captured video to the video hosting platforms and usually are integrated with them. The video platforms allow for manually uploading video produced by camera crew, as well.

The majority of the video hosting platforms are commercial products, therefore, the licensing cost consists a significant amount of whole service cost. Therefore, organizations, especially on education, are moving to adopt [22] open source platforms [20, 23, 45] for several reasons. Minimizing the licensing cost is one reason, a second important reason is that with open source platforms the organizations are more flexible since they keep open the path towards a future transition to other platform contributing thus to the long-term content sustainability. A third reason, applied to universities, is that some open source platforms [7, 32, 45] are designed for the educational sector focusing on the specific needs of the universities instead for business purposes.

The online video usually is not communicated to the end users directly via the video platforms but via Learning Management Systems (LMS), which includes the video links. The LMS communicate with the video platforms forming an ecosystem.

Since the video content may be open, both in terms of access and licenses, some video platforms serve as Open Education Resources repositories, which focus on video, audio, and slides resources. OER repositories provide functionality for uploading, searching, tagging with intellectual property rights metadata, downloading content [13, 38, 42].

## 2.3 Functionality of video platforms

The video platforms provide the functionality of the OER repositories, except the downloading of the video – a feature not desired by the organizations deploying video platforms – and in addition, functionality for processing, managing and publishing the video content. Table 1 presents the main categories of video platforms functionality resulted after a review of popular open source and commercial platforms [20, 34, 45] and OpenDelos.

**Table 1** Main categories of video platform functionality

Category	Short description
Content uploading	Uploading of various content types (video, audio, slides, subtitles) either manually or automatically
Scheduling	Scheduling of live streaming and video recording
Editing	Content and metadata processing such as: edit metadata, trimming/clipping/cutting video, synchronizing video with slides, access rights
Searching	Searching for end users and administrators
Playback	Playback live and recorded content
User management	User account management and roles' administration
Content distribution	Distributing content to third party services and platforms

## 2.4 Functional requirements for the video platform

At the designing phase, the Greek HEIs had set specific functional requirements, which the video platform had to fulfil. Table 2 presents the main requirements for the video platform.

**Table 2** Main functional requirements

Code	Main requirements	Reasoning
R1	Open source platform	<ul style="list-style-type: none"> <li>• No license costs</li> <li>• Future transition to other platform</li> <li>• Fits better to the greek universities model</li> </ul>
R2	Support for live streaming in addition to on demand viewing	Live streaming is a strong requirement by the Greek Universities
R3	Scheduling IP cameras for automatic live streaming, recording and video uploading via a video streaming server	<ul style="list-style-type: none"> <li>• IP cameras provide for a robust and relative low cost for streaming/recording</li> <li>• Video streaming server allows for both recording and distributing incoming streams</li> </ul>
R4	Support for slides uploading and synchronization with the respective video resource	Most the slides were in power point and pdf formats
R5	Support basic online video processing capabilities such as trimming, editing and clipping in with a very friendly user interface	Allows the content administrators and faculty staff to perform basic processing without the need of external software
R6	Metadata support for Creative Commons licensing;	The main aim of the open courses projects is to develop and provide open content. Therefore, there is a specific need to consider the video resources under the perspective of open education resources (OER).
R7	Tailored made metadata to HEIs workflow	<p>Unique presentation of the data that concern the lectures and events e.g., University, Department, Teacher, Course or Event name, which;</p> <ul style="list-style-type: none"> <li>• Avoids data duplication in the metadata;</li> <li>• Facilitates the search/browsing functionality;</li> <li>• Depicts the HEI structure as University, School, and Department.</li> <li>• Relates Teachers, Courses, Classrooms to the Departments.</li> </ul>

The main requirement was to use/implement an open source platform (R1), as discussed in the previous sections. The provision of streaming live lectures (that are part of regular courses in the classroom) and events (R2) contributes in supporting students with the remote attendance capability for cases they cannot attend physically in classrooms. The scenario of exploiting High Definition IP cameras (R3) allows for a sustainable solution, since no camera crew or staff for processing and uploading the videos, is required.

The provision of recorded video, synchronized with the presentation slides if required (R4), satisfies the need of (a) making available online the video from events or regular courses, enabling teachers' reflection and flipped classroom, (b) hosting video that fulfil the demanding requirements driven by MOOCs.

Therefore, video platforms, with the aforementioned functionality, constitute a significant part of the e-learning support ecosystem, for a HEI, along with the Learning Management Systems for regular courses/MOOCs, as well as Educational Resource Aggregators.

None video platform available at the design phase had fulfilled all the requirements and this fact led the development team deciding to develop its own open source video platform OpenDelos. Table 3 presents the functional differences among the video platforms which are valid not only during the design phase (2010) but also nowadays.

The majority of the video platforms are commercial, e.g., Panopto [34] and Kaltura commercial edition [20], which was a major conflict with the requirement about open source platform (R1). A limited number, such as Opencast platform [45], Kaltura community edition [20], are open source platforms. From these, Opencast does not support live streaming (R2) nor the uploading of slides and online synchronization video with slides (R4). For the screen recording, it supports VGA grabbers that record the screen as video. Kaltura's Community Edition platform's functionality is not clear regarding R2 and R3, it does not support tagging content based on Creative Commons licenses (R6) or unique presentation of the data and metadata (R7). The functional requirements are fulfilled by the OpenDelos platform, since it was developed to provide the required functionality. This functionality consists OpenDelos's innovation: an open source platform that supports not only video on demand functionality but also live streaming functionality along with the related scheduling functionality, a platform suitable for HEI since depicts its academic structure, ensures unique data presentation of the elements related to the structure facilitating the searching and browsing features, a platform suitable for hosting open content since metadata that declare the open intellectual copy right licence of each video are available.

**Table 3** Functional differences

Code	Opencast	Kaltura	Panopto	OpenDelos
R1	✓	✓	–	✓
R2	–	? <sup>a</sup>	✓	✓
R3	–	?	✓	✓
R4	–	✓	✓	✓
R5	✓	✓	✓	✓
R6	✓	–	–	✓
R7	–	–	–	✓

<sup>a</sup> “?” means that it is not clear if this feature is supported by the Community Edition (open source)

### 3 OpenDelos design: architecture and implementation

The OpenDelos platform [32] is a rich media and lecture platform for education developed in the context of the aforementioned horizontal project. The platform provides functionality for managing media, such as videos, audios, slides, produced in lectures in the context of regular courses or events.

In addition to the expected functionality of publishing media for on-demand viewing, OpenDelos supports live streaming as well, with a functionality customized to a HEI needs. This innovative functionality distinguishes OpenDelos from other video platforms.

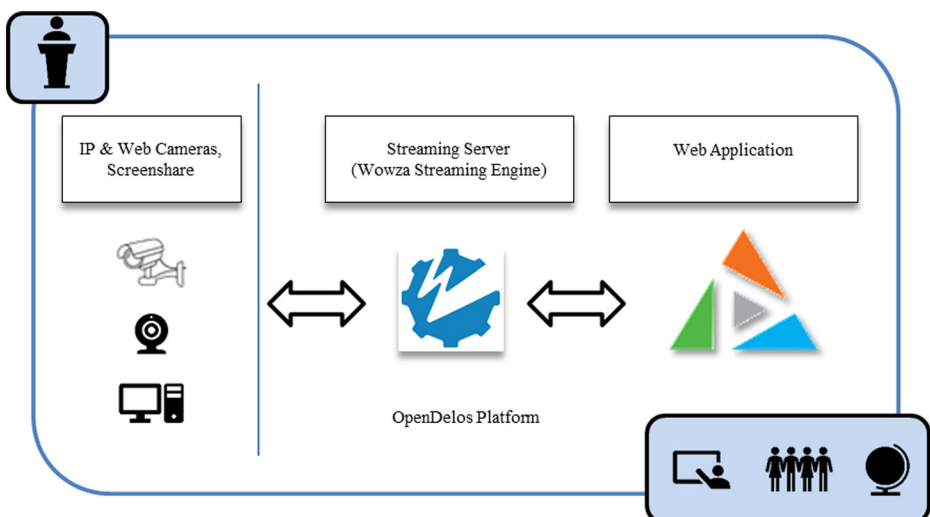
There are 3 major components (Fig. 1) comprising the platform:

- IP cameras
- Streaming server
- OpenDelos Web application, providing integrated functionalities to users

#### 3.1 IP cameras

The OpenDelos platform enables live streaming by (a) either using IP cameras installed in the classrooms for the case of lecturing in a classroom or (b) using web camera for the case of lecturing in office. In addition to the video streaming, slides or screen sharing is also supported. The instructor may either upload its presentation slides before its lecture, or share its screen, in real time, by using an integrated java-based screen sharing application provided along with OpenDelos platform. In any case, the slides presentation or the screen stream is available to the students during live streaming along with the video in a separate frame in the player.

Regarding the IP cameras' infrastructure, most Greek HEI have installed several high definition IP cameras in their teaching rooms and amphitheatres, e.g., 50 such cameras have



**Fig. 1** OpenDelos Platform Components

been installed at National and Kapodistrian University of Athens, Greece and 45 at the Aristotle University of Thessaloniki, Greece. These cameras are used both for live streaming and for the automatic recording of the lectures. The advantages of using IP cameras are (a) there is no need for a camera crew, (b) the publication of the video-lectures is automatic and immediate.

### 3.2 Streaming server

For the recording and live streaming, the OpenDelos collaborates with the Wowza Streaming Engine (WSE) [51], which is the industry leader for live and on-demand streaming. WSE can ingest any live stream and deliver video and audio streams to any player and any device, over any protocol. WSE can record a live stream to a file available for on-demand playback. Most importantly, WSE is programmable, therefore, developers can leverage WSE's comprehensive suite of tools, APIs, and SDKs to automate the required workflows. The streaming server is an external component to OpenDelos platform used mandatory only for the recording and distribution of the live streams and not for the on demand delivery. For the on demand delivery, http pseudostreaming is used as default delivery method whereas rtmp streaming via Apache, NGINX or WSE is also supported. The replacement of the WSE by an open source streaming server, as default server, for supporting recording and distribution of live streams is under investigation.

### 3.3 OpenDelos web application

The web application is mainly written in Java, and it is implemented using the open source Spring Framework [43]. Spring is the most popular framework for building Java Enterprise Edition (EE) applications. It provides model-view-controller architecture and ready components that can be used to develop flexible and loosely coupled, cross-platform web applications. For implementing aspects of the platform, additional modules of the framework were used, namely Spring Data, Spring Security and Spring Web Services. An Apache Tomcat server installation is required to deploy the application. Tomcat is often integrated with Apache server to enhance Apache with servlet capabilities and to capitalize on Apache's optimized and robust static-page delivery mechanisms. As an alternative, tomcat server, is integrated with NGINX server as well. OpenDelos supports both integrations.

The main goal of the application is to support the creation and sharing of educational resources, named videolectures, in a HEI. A video-lecture is an autonomous educational resource consisting of a recorded lecture together with an intimate set of metadata describing its source, creator (owner), editor, speaker(s), topic, date and time created, (intellectual) rights held in and over the resource, access URL and video file properties.

“Creators”, are mainly the faculty members of an educational institution. OpenDelos, also defines groups of administrators that are granted editing rights on resources on behalf of one or more faculty members. These groups include institution, school, department administrators and support personnel that are granted specifically defined rights and range of authority over content (i.e., administrators can edit resources created in their respective administrative unit). Faculty members have complete control over editors' rights, in lecture level. Any person involved in the creation and editing of a videolecture, is clearly recorded and it can be uniquely identified, in the videolecture's metadata. Users with editing or creation rights are described as “content managers” throughout this document. Spring Security modules are used to identity and authenticate all users, in collaboration with the institution's Single Sign-On server. In

current version of OpenDelos, LDAP [24], CAS [8] and Shibboleth [40] authentication servers are supported. Users of administration groups can also be created and stored locally. Using Spring Security libraries, OpenDelos can handle all aspects of security, including comprehensive and extensible support for authentication, authorization and data protection.

“Source” defines the administrative unit and course in the framework of the institution’s curriculum.

“Topic” is represented using keywords, key phrases, or free text, in the elements: title, description, category and keywords. “Category” uses a controlled vocabulary.

“Intellectual rights” are selected by the creator during the editing process, using the Creative Commons family of open licenses. Before publishing a resource a creator must choose which rights, if any, are held for the resource ranging from the default “Attribution” to “Copyrighted Material”.

“Video properties” include basics information about the video file itself like format, size, aspect ratio and duration.

The web application itself is comprised by five major modules: (a) “Institutional Data”, (b) “Scheduler”, (c) “Media Resources”, (d) “Search” and (e) “Publishing and Third-Party” modules. These modules are depicted in Fig. 2 and presented in the following paragraphs.

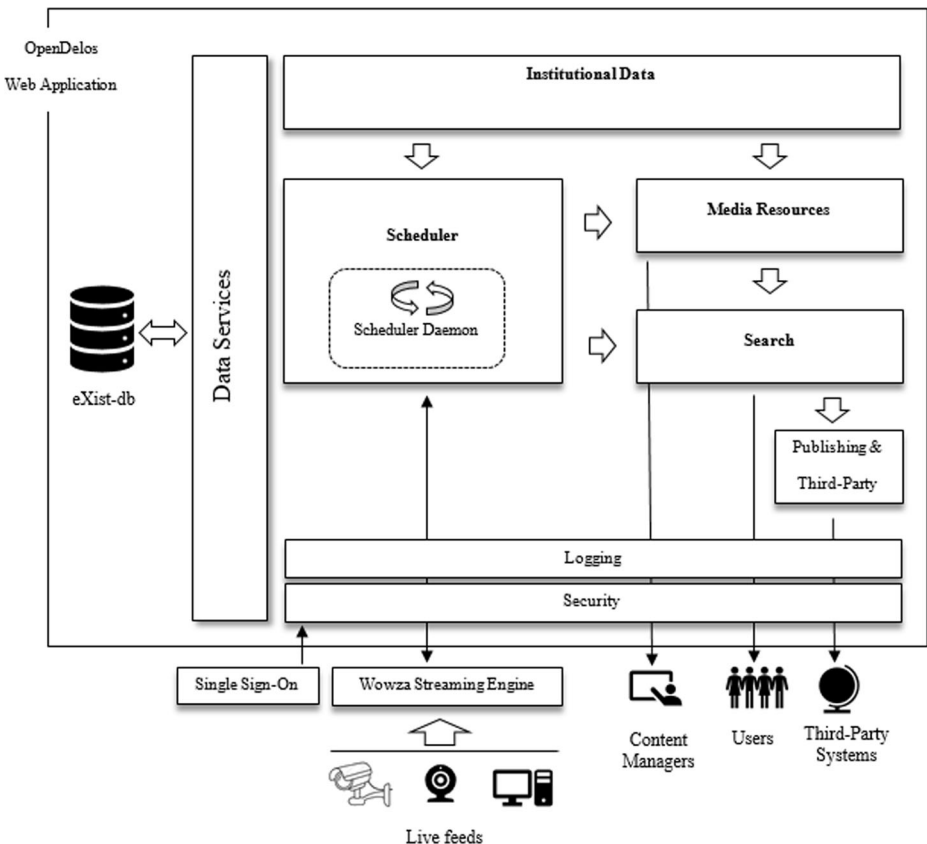


Fig. 2 OpenDelos Web Application Architecture

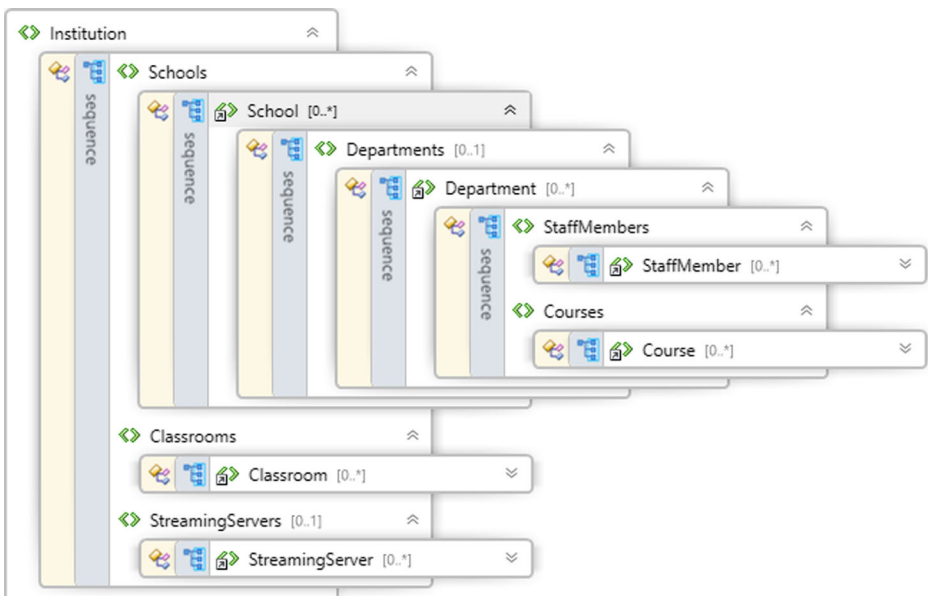


### 3.3.1 The “Institutional data module”

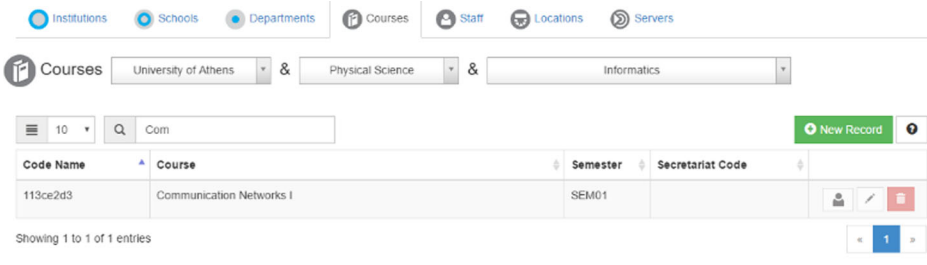
This module allows for inserting accurate information about the institute’s structure (Fig. 3), e.g., schools, departments, teachers, courses, classrooms, servers used etc. The accurate information facilitates (a) the metadata entry by the Scheduler and Media Resources Modules (b) the content searching and browsing by displaying the information uniquely avoiding the annoying multiple display of similar information, e.g., same information with capital letters or typos errors, etc. Institutional data are made available to all other modules of the platform through the service layer of the web application, either by Java API calls or by utilizing RESTful Web Services [36].

Representing and storing the institute’s structure in a database was a challenge. At first, a traditional relational database was considered, but very soon, the development team has abandoned the idea due to the data nature of the model. The highly hierarchical and deeply nested structure of the data (Figs. 3 and 4), would eventually make relational schemas very complex to manage and difficult to query. The Object-Relational mapping analysis showed that we needed 52 Tables to store the institute’s structure. The same was true also for the other primary domain object of the platform, the “Lecture”, which was designed to represent a video-lecture resource. In addition, a basic requirement for the platform, demanded from OpenDelos to be ready to share metadata information with third-party systems, by incorporating the Simple Dublin Core standard [14], which describes digital web resources using XML. XML is very well suited to deeply nested data and mixed content. It is also highly scalable. We decided that XML was perfect for our case.

The XSD schemas required relatively small effort to design and the domain objects are stored by using only one (1) easy manageable XML file per instance. Spring’s direct support to JAXB 2.0 was available to carry the mapping of XML files to domain objects and vice-versa.



**Fig. 3** Institution Schema (attributes omitted)



**Fig. 4** Example of the Institutional Data module user interface

To support this decision, eXist-db database [52] was inherited. Exist-db is an open source Native XML Database written in Java, which allows software developers to persist XML documents without writing extensive middleware. It is easy to use, optimize and index data. Exist-db follows and extends many W3C XML standards such as XQuery and XPath for querying and editing XML documents, and includes a powerful document search engine, based on Apache Lucene [2].

### 3.3.2 The “scheduler module”

The “Scheduler Module” is the most sophisticated module of the platform. It enables the scheduling of the live and recording sessions in strong collaboration with Wowza Streaming Engine. It was implemented to serve the following use case (Fig. 6):

1. *Content manager adds the schedule for a course or event.*

It can be a reoccurring or one time session. S/he can use any available teaching room i.e. fixed IP camera or a personal web camera. S/he can also upload a presentation file to accompany his/her live session or choose to share a computer’s screen during the broadcast. All live session can be recorded. It’s in the user’s discretion to record a session but recording it’s encouraged.

2. *A message is sent to the streaming server to start live broadcast and/or recording of the specified stream, when the time comes.*

Recordings are saved to a pre-configured shared storage.

3. *(Live sessions) create videolecture live-entry and publish session details, through the search module.*

Live broadcasts are accessible (freely or password protected) to end users through the web application’s Search module. A rich media player is used to playback the live content along with the optional screen share or presentation. A user can actively browse through the presentation’s slides.

4. *A message is sent to the Streaming Server to stop live broadcast and/or recording, when the time comes.*

5. (Recorded sessions) Gather the recorded file(s) and create the on-demand version of the videolecture
6. (Optional) Publish the videolecture.

Following the user’s selection, a videolecture be published immediately or set to a private state. Authorized users, can on a later time, edit the video file and metadata using the tools and editors of the “Resources Module” (discussed below) and (re) publish the resource. All published resources are available to end-users and third-party systems, through the application’s “Search” and “Publishing & Third-Party Sharing” modules (discussed below).

The module’s front-end user interface (Fig. 5) accommodates the scheduling of live and recorded sessions. Behind the scenes, the module’s services are responsible for several tasks, including: preventing overlapping sessions in the same teaching room, calculating dates of reoccurring sessions throughout the active period, exclude holidays, fill live and on-Demand videolecture entries with the minimum required metadata (teacher, course, category, license) etc.

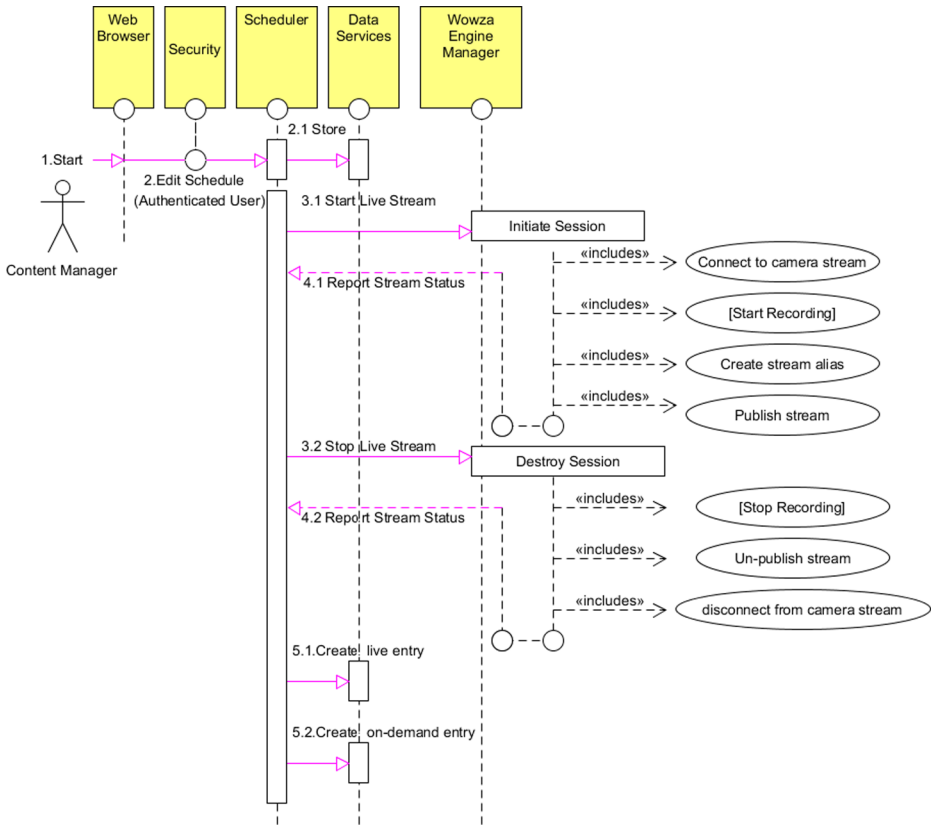
There are two implementations available in OpenDelos platform, that can be used to support the communication between this module and the Streaming Server.

The first (Script Daemon), is implemented as a middleware service between the Scheduler module and the Streaming Server. It’s series of shell scripts which are maintained locally on the streaming server. The service reads the timetable from the Scheduler in pre-defined time intervals, and sends corresponding start/stop stream messages (commands) to the Streaming Server, calling RESTful endpoints, using Unix-based “curl” command. The communication between the Scheduler Module and the Streaming Server is initiated at specific time intervals, by programming the Unix-based “cron” job scheduler utility. This solution is best suited for older version of WSE (version  $\leq 4.5$ ) and can be easily modified to support other streaming engines in the future (Fig. 6).

New lecture

<p><b>Institution:</b> <input type="text" value="University of Athens"/></p> <p><b>Department:</b> <input type="text" value="Informatics"/></p> <p><b>Course:</b> <input type="text" value="Communication Networks I"/></p> <p><b>Speaker:</b> <input type="text" value="Merakos Lazaros"/> <small>(Prof. Informatics)</small></p> <p><b>Additional information:</b> <input type="text"/></p> <p><b>Room:</b> <input type="text" value="Panepistipolis"/></p> <p><b>Type:</b> <input type="text" value="Regular"/></p> <p><b>Period:</b> <input type="text" value="Winter term"/></p> <p style="text-align: center; font-size: small;">Period 26/9/16 - 31/1/17</p>	<p><b>Day:</b> <input type="text" value="Tuesday"/></p> <p><b>Hour:</b> <input type="text" value="12:00"/></p> <p><b>Duration:</b> Hour: <input type="text" value="2"/> Minutes: <input type="text" value="0"/></p> <p><b>Transmission:</b> <input type="text" value="Transmission"/></p> <p><b>Access type:</b> <input type="text" value="Institutional account"/></p> <p><b>Recording:</b> <input type="text" value="Automatic"/></p> <p><b>Publishing policy:</b> <input type="text" value="Private (approval req...)"/></p> <p><b>Additional Resources:</b> <input type="text" value="Screen sharing"/></p>
---	---

Fig. 5 New entry addition to the scheduler



**Fig. 6** Scheduler Use Case Diagram

The second solution (Integrated Daemon), is integrated into the implementation of the Scheduler Module. It performs the exact same tasks, but without the need for any external services.

The functionality of the “cron” is replaced within the module by using the integrated Spring’s Quartz scheduling library support. The curl command has been replaced with direct HTTP GET and POST requests from the Scheduler Module to Wowza Streaming Server HTTP Providers, which are Java classes that are configured on a per-virtual host basis. They are lightweight HTTP servers that can be used to query or interact with the Wowza server. The shell scripts business logic has been replaced with a custom made Wowza Server Module. Wowza Server Module is also a set of Java classes that provide much of the functionality needed to control the streaming process from a remote location. Integrated Daemon requires the newest Wowza Java API available in the latest versions of WSE (version >4.5). Using the latter implementation of the Daemon, eliminates the need of Unix-based commands, thus, making OpenDelos entirely cross-platform.

Communication and session identification between the Scheduler Module and the Streaming Server, is based on an identity key calculated for each session. This is the structure of this session identity:

*InstitutionId.DepartmentId.TeachingRoomId.CourseOrEventId.TeacherId.DateTime*

where:

InstitutionId:	Institution identity	e.g. uoa
DepartmentId:	Department identity	e.g. math
TeachingRoomId:	Teaching Room identity	e.g. mathlab1a
CourseOrEventId:	Course of Event identity	e.g. math100
TeacherId:	Teacher identity	e.g. staffmember23
DateTime:	Start date and time of Session	e.g. 28.01.2017–14:00

This session identity is both unique and descriptive for every session. It comprised from the identity of each data that jointly identify each session, including date and time, separated by a period character (.).

When a stream is initiated, the Daemon is responsible for locating the corresponding IP camera stream that matches the teaching room, extracted by querying a hard-coded mapping table, stored on the streaming server, with the session id (Teaching Room  $\leftarrow$   $\rightarrow$  URL).

It also has the “ability” to hide the actual camera stream URL from the end user. The latter is accomplished in two different ways:

The Script Daemon, uses a hashed version of the session id, encrypted with a shared secret both known to the Scheduler Module and Streaming Engine. The hashed key is used to create an alias URL, hiding the actual URL. Onwards, since the session id is unique for each session, the alias URL becomes invalid after the session ends.

The Integrated Daemon, used a little different approach. The session id is enclosed in a JSON Web Token (JWT) [19] encrypted also with a shared secret. The JWT is sent to the Streaming Server as a parameter to the HTTP POST request when it’s time to start a live session. The string representation of the token is used to create an alias URL hiding the actual stream URL. The token’s validity duration matches the duration of the corresponding session. This way, the token and thus the alias URL, expire when the session ends and cannot be re-used. In both versions, the alias functionality is currently provided by inherited Wowza’s StreamName Alias module.

When a recording stream finishes, a message is sent to the Scheduler Module to create the lecture’s offline version. The session’s id helps in filling the session’s minimum metadata set. Teacher’s profile also supports this task. The session id also predicts the name of the recorded file in the shared storage area which is moved to a configurable permanent folder, from where it can be served on-demand. It’s a matter of preference and configuration if the on-demand files are served by the streaming server or with pseudo-streaming by the Apache/NGINX web server. OpenDelos supports both scenarios. To better handle the files in matter of size, long recordings are split based on duration. Therefore, a session of 3 h will produce by default 3 files of 1-h each and consequently 3 offline lecture entries, marked with appropriate tags (i.e. part 1, part2, part3). The duration of each split is configurable. The split service is provided by the Wowza Streaming Engine.

### 3.3.3 The “media resources module”

The “Media Resource Module” provides the tools for editing metadata and multimedia for the on-Demand Services. In addition to the automatic video/audio feeding from the Streaming Server, OpenDelos supports the uploading of video or audio files, slides, subtitles via the Resources Module.

The content manager may upload video produced by a camera crew, in mpeg-4 format, or instead of video, audio files in mp3 format. File uploading implementation uses Plupload [35], a JavaScript API for building file uploaders. Among its many unique features, Plupload supports chunked uploads, indicated for accepting large file uploads, such as video files, overcoming the HTTP protocol upload limit, web server limits and most likely, hard to predict, browser timeouts. Also, Plupload supports HTML5 and when necessary can fallback to alternative runtimes, like Flash and Silverlight. Video/Audio files are analyzed immediately after uploading with FFMPEG [15] to extract format, size, duration, resolution and aspect-ratio.

Users may also upload the slides of a presentation. File formats that are currently supported are the following:

- Microsoft's PowerPoint Presentations (ppt and pptx)
- Adobe PDF files
- JPG Images (mass image posts can be uploaded compressed in a ZIP file and automatically extracted on server).

The presentation file is converted to a series of images, using the Apache POI [4], an open source Java library for Microsoft documents or the Apache PDFBox [3], an open source Java library for working with PDF documents. All extracted/uploaded images are edited to a configurable resolution and aspect-ratio. Header texts from slides are extracted and indexed to assist the search functions.

Finally, the content manager may upload subtitles, in many languages, in SRT format.

Apart from the media files, the content manager should enter the metadata that concerns the video/audio file (Fig. 7). These metadata are required for tagging the video and facilitating the search by the end users and content managers.

The Media Resources Module provides the content managers with the capabilities of editing the metadata of the resources and applying, online, basic video processing operations, such as virtual and real video trimming and editing (Fig. 8) and synchronizing video with the uploaded slides (Fig. 9). The user interface of these online tools was designed to be intuitive and very friendly to be used by people with no technical background. Both tools are implemented solely with JavaScript (and JQuery). Requests to the underlying web application for getting and saving editing data are achieved with ajax calls to RESTful Web Services.

### 3.3.4 The “search” module

This module is responsible of displaying searchable lists of:

- Live Resources
- On-Demand Resources
- Programme of scheduled live broadcasts for the active period

For the end-user, a simple, friendly and intuitive user interface has been designed. The first action for the end user is to select if s/he wishes to search for uploaded videos/audios, watch live streams or check the schedule for future broadcastings.

S/he - may search - available content by providing key-words or browsing it based on several predefined criteria: the topics categories list, courses list, events or teacher list.

Course Lecture Multimedia Presentation Subtitles

**Title \*:**

**Description \*:**

**Speakers \*:**

**Category \*:**

**Keywords:**   
*Press 'Enter' after each word*

**Series:**

**Date & Time:**

**License:**

**Player Options:**  Play Introduction Video  
 Show Legend

**Language:**

Fig. 7 Inserting metadata along with multimedia (video, audio) uploading

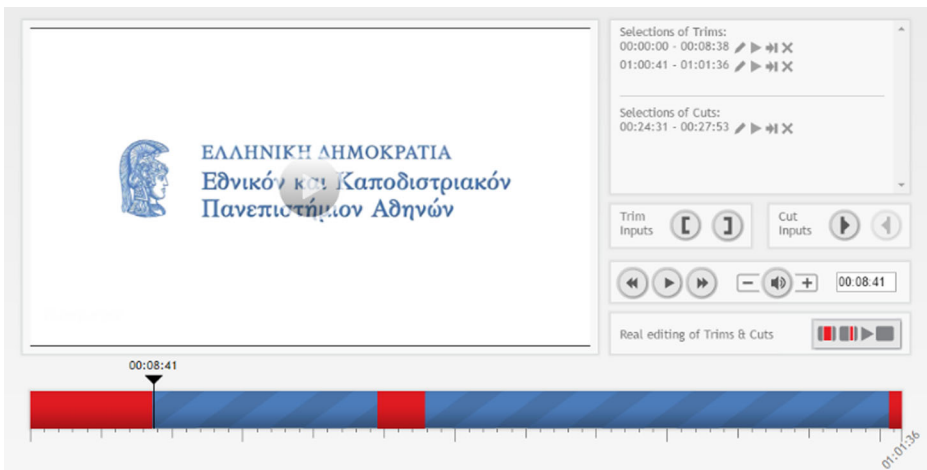


Fig. 8 OpenDelos online video editing

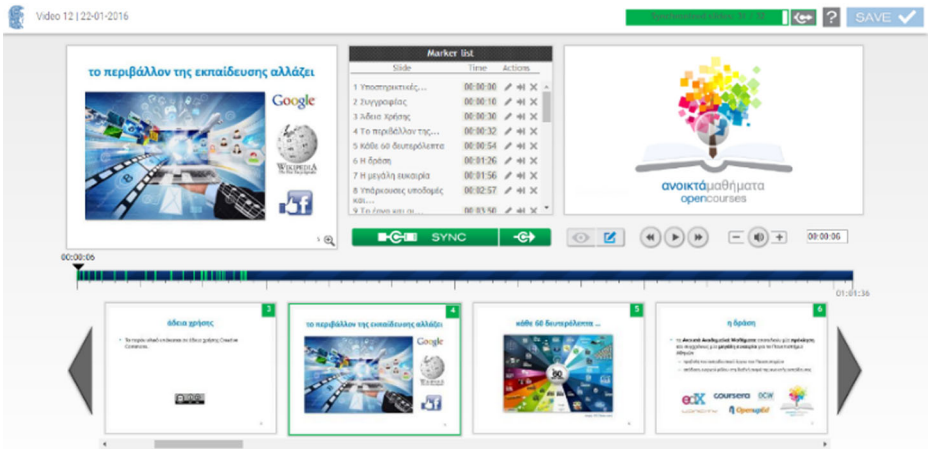


Fig. 9 OpenDelos online video & slides synchronizing

The search or browsing results that fulfil the search keyword or the browsing criteria appear. In the sequel, the user may apply further filtering to the results based on the rest of the aforementioned predefined criteria (topic category, course, event, person, year, etc.).

The same search functionality is available to the content managers via the Media Resource Module, who may perform edit and processing actions on the resources that fulfil the search or browsing criteria.

3.3.5 The “publishing and third-party sharing” module

The OpenDelos platform enables publishing live streaming and recordings for on-demand viewing by its responsive Rich Media Player, which is capable of playing the content at desktop devices or tablets (the latter only on demand) (Fig. 10).



Fig. 10 OpenDelos responsive player



In addition, RSS feeds are also available for integration with other systems as well as the option of automatically publishing to iTunes and YouTube.

All publicly available metadata, including institution structure and scheduling information, stored in the database are available through RESTful Web Services [36], which allows the development team or other developers to easily create tools and services to collaborate with third-party systems. The platform can also share private metadata with trusted clients, utilizing JSON Web Token (RFC 7519 standard) where authentication and authorization is required. As a proof of concept, OpenDelos shares metadata information with NOCS portal by exploiting the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) [31] and provides access to searchable multimedia content to Open eClass LMS through published RESTful Web Services. All services are exposed as Web Services, exchange data formatted in either JSON or XML on demand. End user search result in stored or live content or scheduled events are also made available through RSS and ATOM feeds.

### 3.3.6 Other aspects

Wide spread technologies and de-facto standards were used to design and implement the user interface of the web application including: Apache tiles, JSP, HTML, CSS, Bootstrap, etc.

Literal texts throughout the application are stored in separate resource files, allowing the platform to be installed in any language, by simply providing a line-by-line translation of the resources files to the desired language. OpenDelos provides Greek and English resource files out of the box.

### 3.3.7 Accessibility evaluation

All end-user pages follow the Web Content Accessibility Guidelines (WCAG) 2.0 [50], which cover a wide range of recommendations for making Web content more accessible. Specifically, the OpenDelos platform has been evaluated regarding its compliance to the WCAG 2.0 and Section 508 Guidelines [39]. The evaluation tools used were WAVE [49], AChecker [1], and T.A.W. [44]. Additionally, the software tool ColorBlind Web Page Filter [10] was used to evaluate the brightness and contrast foreground and background colours for text elements.

The accessibility evaluation was an iterative process carried out by external experts from Accessibility Unit for Students with Disabilities of the National and Kapodistrian University of Athens.

## 4 Heuristic evaluation

This section presents the methodology, the profile of participants and the evaluation results concerning OpenDelos heuristic evaluation.

### 4.1 Methodology

In the present study, the inspection method called heuristic evaluation was chosen for usability testing. This method is a discount usability testing method in which a small group of experts evaluate (by means of general heuristics – rules of thumb) whether the interface of a system is in accordance with recognised usability principles [29]. The experts use their experience and

knowledge from usability to evaluate a system under development (mock-ups) or an existing system [25]. According to [25], the participation of five experts in a properly conducted usability test will usually ensure detection of approximately 80% of all usability problems. Furthermore, and according to [28, 48], it is impossible to detect all problems super users may encounter.

In our case we have recruited six (6) experts. Before the evaluation, an hourly experts' meeting held for a brief presentation of OpenDelos platform to the experts. During this meeting, both the heuristics and the severity classification of problems were briefly discussed, to ensure that they were understandable. After the meeting, two files have been sent by e-mail to the experts: (a) a brief description of the heuristic evaluation process, and (b) a template for marking their evaluation and writing down their comments for every heuristic evaluation rule.

The heuristic evaluation has been conducted in two phases. The first phase was a quick visit to every screen of the entire OpenDelos application, in order to inspect the flow UI dialogs. In the second phase, the focus was on every individual item in each screen.

The experts evaluated the OpenDelos individually in order to avoid bystander bias and to ensure an independent assessment. The experts evaluated the system through constructed scenarios and judged its compliance with Jakob Nielsen's ten general usability principles for interaction design ([26, 30]). They continuously evaluated the various dialogue elements of the system and compared them with the heuristics. Usability problems were categorised according to Jakob Nielsen's severity scale [26]: 0 = I don't agree that this is a usability problem at all. 1 = Cosmetic problem only: need not be fixed unless extra time is available on project. 2 = Minor usability problem: fixing this should be given low priority. 3 = Major usability problem: important to fix, therefore should be given high priority. 4 = Usability catastrophe: imperative to fix this before product can be released. The frequency of a usability problem has been filled according to the following scale: 0 = Rarely. 1 = In a few dialogs/screens. 2 = In some dialogs/screens. 3 = In many dialogs/screens. 4 = In very many dialogs/screens.

A table was made, which contained the ten usability principles and columns with problems, severity, frequency, and solutions that experts could easily fill out. Some of the problems from the list were considered more important than others; and by using the severity scale, it would be possible to prioritise problems for a future redesign process. Some of these problems were considered more frequent than others; and by using the frequency scale, it would be also possible to prioritise problems for a future redesign process. Each of the six experts used about three hours for testing. During the testing they have completed the template administered to them for each interface (a) basic user interface and (b) admin interface. The features of OpenDelos platform evaluated were the following: (a) Basic User Interface (e.g. learner): (i) Search educational videos and (ii) Watch educational videos; (b) Admin User Interface: (i) Post a video from a classroom; (ii) Post a video from an event; (iii) Enter/Edit video data; (iv) Enter/Edit event data; (v) Search content; (vii) Insert user.

## 4.2 Participants

OpenDelos was tested by six (6) experts (five men, one woman). The experts were computer engineers or computer scientists, having MSc degree or PhD diploma in Information and Communication Technologies in Education. They are considered to be usability as well as interface experts [29]. They were recruited from another University in Greece and worked as PhD students ( $n = 5$ ) or assistant professor ( $n = 1$ ). The average age of the 6 experts was 38 years (min 28, max 55). All experts received verbal and written information about the

purpose and the method of the study, and were also informed that they had to take into account that the OpenDelos system had been developed for users having basic knowledge in multi-media editing and massive open online courses support. Table 4 features the experts' characteristics.

### 4.3 Evaluation results

For the evaluation, each expert ticked off the relevant columns and cells in a table featuring the ten heuristics, described the problems encountered, and graded the severity and the frequency of the problems. The problems and notes from the evaluations were structured and analysed. The data were described in a simple bar chart to compare the number of problems among experts and their severity.

#### 4.3.1 Opendelos basic user interface

The experts identified a total of 68 problems in the OpenDelos basic user interface of which 18 (26.5%) were unique problems (unique refers to a problem that could be repeated in more than one place but still counted as a single, unique problem with several instances). Each expert identified 8–14 problems. Two problems (2.9%) were found by all experts. These problems were related to frequently asked questions, absence of help files.

The features of OpenDelos platform evaluated by the experts concerning the basic user interface were: (i) Search educational videos and (ii) Watch educational videos. Concerning searching of educational videos, the experts have not identified any problem concerning heuristic principles 1, 4, 6, and 8. Most of the problems of this feature concern the integration of help and the correction of search forms. Concerning video watch, the experts have not identified any problem concerning heuristic principles 1, 2, 3, 5, 8, and 9. Most of the problems of this feature concern the integration of help and the usage of metadata.

Table 5 provides an overview of the experts' classification of the severity of the problems found at the OpenDelos basic user interface. The most frequent classification was "Severity 3" (55.8%) with 38 of the problems.

Figure 11 illustrates the distribution of problems of the OpenDelos basic user interface into heuristics. Heuristics 3, 7 and 10 were not used, but the three most frequently used heuristics

**Table 4** Experts' characteristics

	Sex	Age	Education	Profession
Expert 1	Male	40	MSc in Computer Engineering and Informatics. PhD in Information and Communication Technologies in Education	Assistant professor
Expert 2	Female	45	MSc in Informatics (Information and Communication Technologies in Education)	PhD fellow
Expert 3	Male	28	MSc in Informatics (Information and Communication Technologies in Education)	PhD fellow
Expert 4	Male	30	MSc in Informatics (Information and Communication Technologies in Education)	PhD fellow
Expert 5	Male	55	MSc in Informatics (Information and Communication Technologies in Education)	PhD fellow
Expert 6	Male	30	MSc in Informatics (Information and Communication Technologies in Education)	PhD fellow

**Table 5** Number of problems of the OpenDelos basic user interface identified and their severity classification according to expert

	Severity 1	Severity 2	Severity 3	Severity 4	Total
Expert 1		1	8		9
Expert 2			8		8
Expert 3		1	13		14
Expert 4	8	5			13
Expert 5		5	9		14
Expert 6	8	2			10
Total	16	14	38	0	68

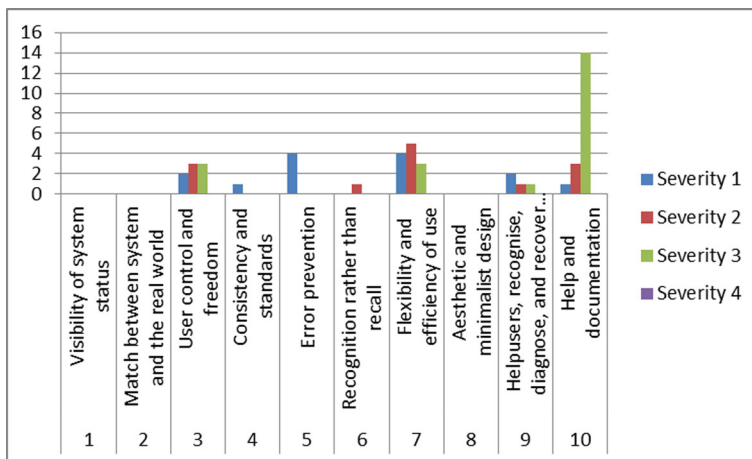
were: “Help and documentation” (38%) with 18 of the problems, “Flexibility and efficiency of use” (25%) with 12 of the problems, and “User control and freedom” (17%) with 8 of the problems. The problems identified in “Help and documentation” were related in particular to the absence of help files and documentation. Within the heuristic, “User control and freedom”, there comments about the usage of specific buttons and icons without their meaning.

#### 4.3.2 OpenDelos administrator user interface

The experts identified a total of 228 problems in the OpenDelos administrator user interface of which 61 (27%) were unique problems. Each expert identified 31–44 problems. Eight problems (4%) were found by all experts. These problems were related to frequently asked questions, absence of help files.

The features of OpenDelos platform evaluated by the expert3s concerning the administrator user interface were: (i) Post a video from a classroom; (ii) Post a video from an event; (iii) Enter/Edit video data; (iv) Enter/Edit event data; (v) Search content; (vii) Insert user. The experts have not identified any problem concerning the following heuristic principles:

- 1 and 9 for the feature “Post a video from a classroom”.
- 1 and 8 for the feature “Post a video from an event”.

**Fig. 11** Problems of the OpenDelos basic user interface identified on the basis of Nielsen's ten usability principles and classified according to severity

- 1 and 8 for the feature “Enter/Edit video data”.
- 1, 5, 6, 7 and 8 for the feature “Enter/Edit event data”.
- 1, 2, 4, 5 and 8 for the feature “Search content”.
- 1, 2, 4, 5 and 8 for the feature “Insert user”.

Most of the problems of the administrator user interface for the feature:

- “Post a video from a classroom” concern the integration of help and the usage of easy to understand tags in the interface.
- “Post a video from an event” concern the integration of help and the clarity of the interface
- “Enter/Edit video data” concern the integration of help and the clarity of the interface and the usage of web forms.
- “Enter/Edit event data” concern the integration of help and the clarity of the interface and the usage of web forms.
- “Search content” concern the integration of help a and the usage of auto-correct features in web forms.
- “Insert user” concern the integration of help.

Table 6 provides an overview of the experts’ classification of the severity of the problems found at the OpenDelos administrator user interface. The most frequent classifications was “Severity 2” (57%) with 130 of the problems.

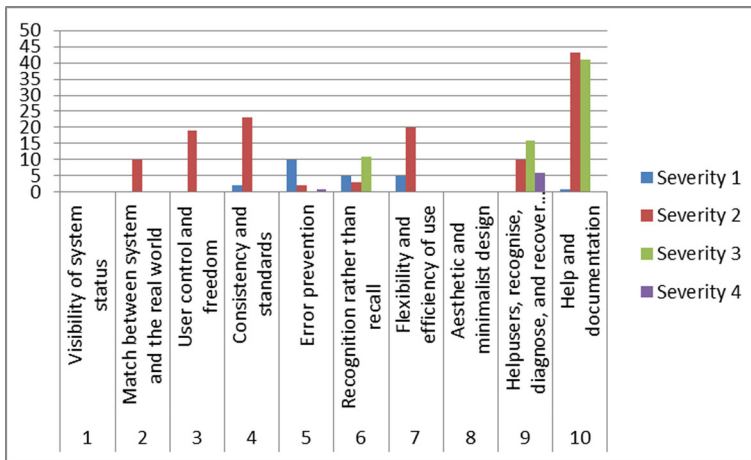
Figure 12 illustrates the distribution of problems of the OpenDelos administrator user interface into heuristics.

#### 4.4 Discussion

According to the results presented above, it seems that the heuristic method is feasible for evaluating the usability of OpenDelos interfaces. The OpenDelos system corrected and the new version used in order to measure the end users acceptance. The goal of the present heuristic evaluation was to discover usability problems in OpenDelos so that these problems can be solved and end-users patients easily can assess the system. In order for OpenDelos to be successful, it was highly important to develop and test the double nature of the system: (a) the basic end user interface and (b) the administrator interface. For the end users OpenDelos could be a platform for accessing educational content and supporting in a intuitive, flexible and usable way their educational process. For the administrators, OpenDelos should be a simple

**Table 6** Number of problems of the OpenDelos administrator user interface identified and their severity classification according to expert

	Severity 1	Severity 2	Severity 3	Severity 4	Total
Expert 1	4	24	16	0	44
Expert 2	4	24	16	0	44
Expert 3	4	31	6	1	42
Expert 4	0	20	10	6	36
Expert 5	10	11	10	0	31
Expert 6	1	20	10	0	31
Total	23	130	68	7	228



**Fig. 12** Problems of the OpenDelos administrator user interface identified on the basis of Nielsen’s ten usability principles and classified according to severity

but effective tool for producing and sharing educational video quickly. The heuristic evaluation was performed using a template in which each expert organised the problems identified according to their severity and offered suggestions for solutions. As expected, the method identified problems of the interface’s nature and it was found to be an effective method for detecting problems in OpenDelos. The identified problems were easy to understand and resolve for technical personnel and education professionals and also readily applicable for the redesign process. The efficiency of this method makes it particularly well suited for iterative design processes. The prepared list of usability problems allowed the experts to quickly identify a large number of problems ( $N1 = 68$  for the basic user interface and  $N2 = 228$  for the admin user interface). Even though most of the problems were classified as “Severity 2” for the admin UI and as “Severity 3” for the basic UI, it is still relevant to find and fix these problems.

The experts felt that the system was quite finished but not well documented. They believe that the OpenDelos environment is beautiful, simple, clean and pleasant. The various forms of categorization is useful and helpful for those who use it. It is noteworthy that no problem has been found by the experts concerning heuristic principles 1 (i.e. “Visibility of system status”) and 8 (i.e. “Aesthetic and minimalist design”) during the heuristic evaluation by experts. The findings indicate that the OpenDelos system did not present major usability problems. Its interface needs slight improvement and the integration of a complete documentation, in order to be deployed easily.

## 5 Conclusions and next steps

The need for video-based learning and the respective video platforms is increasing. The OpenDelos is an open source platform that provides innovative functionality, customized to Higher Education Institutes’ needs and workflows, with a friendly and intuitive user interface. Twenty-one Greek HEIs have already installed OpenDelos and use it in productive mode. This fact shows that OpenDelos’ degree of maturity is high.

OpenDelos is a unique platform satisfying the following requirements for HEIs: (R1) Open source platform; (R2) Support for live streaming and on demand viewing; (R3) Scheduling of IP cameras for automatic live streaming, recording and video uploading via a video streaming server; (R4) Support for slides uploading and synchronization with the respective video resource; (R5) Support basic online video processing capabilities such as trimming, editing and clipping in with a very friendly user interface; (R6) Metadata support for Creative Commons licensing; (R7) Tailored made metadata to HEIs workflow.

None of the available video platforms at the design phase had fulfilled all the above requirements and this fact led the development team deciding to develop and evaluate its own open source video platform OpenDelos. The heuristic evolution showed that the OpenDelos did not present major usability problems. Its interface needs slight improvement and the integration of a complete documentation, in order to be deployed easily. The results and experts' comments have been taken into account and the improved version has been distributed to the end users in order to measure end users' acceptance towards OpenDelos using Technology Acceptance Model (TAM) [12]. First results, with 34 end-users, are encouraging about the perceived ease of use, the attitude towards using and the behavior intention of OpenDelos.

One of our next steps is to apply this evaluation with more participants, in order to extend the sample size significantly. This will help us to get more significant results.

The next steps, regarding OpenDelos' development, are a) to enrich the provided functionality, e.g., enabling interaction with the end users, etc., and b) to design and implement an ecosystem of platforms that collaborate with each other, having OpenDelos as a core component of the ecosystem's functionality.

**Acknowledgements** We would like to thank the end users as well as the HCI experts for their support in the evaluation of OpenDelos platform.

## References

1. AChecker (2017) <http://www.achecker.ca/checker>. Accessed 12 April 2017
2. Apache Lucene (2017) <https://lucene.apache.org/core/>. Accessed 12 April 2017
3. Apache PDFBox (2017) <https://pdfbox.apache.org/>. Accessed 12 April 2017
4. Apache POI (2017) <https://poi.apache.org/>. Accessed 12 April 2017
5. Barokas J, Ketterl M, Brooks C, Greer J (2010) Lecture capture: student perceptions, expectations, and behaviors. In J. Sanchez, K. Zhang (eds) Digital media, p 1A8
6. Bento A (2000) Developing a class session using audio and video streaming. In: Khosrowpour M (ed) Web-based learning and teaching technologies: opportunities and challenges. Idea Group Publishing, Hershey
7. Brooks CA, Ketterl M, Hochman A, Holtzman J, Stern J, Wunden T, Amundson K, Logan G, Lui K, McKenzie A, Meyer D (2011) OpenCast Matterhorn 1.1: reaching new heights. In proceedings of the 19th ACM international conference on multimedia 2011 Nov 28 (pp. 703-706). ACM
8. Central Authentication Service (CAS) (2017) <https://wiki.jasig.org/display/CAS/Home>. Accessed 12 April 2017
9. Choi HJ, Johnson SD (2005) The effect of context-based video instruction on learning and motivation in online courses. *Am J Dist Educ* 19(4):215–227
10. ColorBlind Web Page Filter (2017) <https://www.toptal.com/designers/colorfilter>. Accessed 12 April 2017
11. Daradoumis T, Bassi R, Xhafa F, Caballé S (2013) A review on massive e-learning (MOOC) design, delivery and assessment. In P2P, parallel, grid, cloud and internet computing (3PGCIC), 2013 eighth international conference on 2013 Oct 28 (pp. 208-213). IEEE
12. Davis FD, Bagozzi RP, Warshaw PR (1989) User acceptance of computer technology: a comparison of two theoretical models. *Manag Sci* 35(8):982–1003



13. Doulamis ND, Psaltis C, Doulamis AD, Sotiriou S, Sotiriou M (2008) Technical infrastructure of the COSMOS portal. *iJET* 3(S3):21–32
14. Dublin Core: Metadata Initiative (DCMI) (2017) <http://dublincore.org/>. Accessed 12 April 2017
15. FFMPEG (2017) <https://ffmpeg.org/>. Accessed 12 April 2017
16. Giannakos MN (2013) Exploring the video-based learning research: a review of the literature. *Br J Educ Technol* 44(6):E191–E195.
17. Greek Academic Network (GUnet) (2017) <http://www.gunet.gr>. Accessed 12 April 2017
18. Greek National Open Courses Search portal (2017), <http://opencourses.gr>. Accessed 12 April 2017
19. JSON Web Token (JWT) (2017) <https://jwt.io/>. Accessed 12 April 2017
20. Kultura Community Edition (2017) <http://www.kultura.org>. Accessed 12 April 2017
21. Kay RH (2012) Exploring the use of video podcasts in education: a comprehensive review of the literature. *Comput Hum Behav* 28(3):820–831
22. Klefodimos A, Evangelidis G (2016) Using open source technologies and open internet resources for building an interactive video based learning environment that supports learning analytics. *Smart Learning Environments* 3(1):1–23
23. Kopf S, Wendel V, Zhang L, Atrey P, Timmerer C, Cesar P, Lux M, Griwodz C (2013). SIGMM Records. Association for computing machinery, special interest group on multimedia 5(3). <http://heim.ifi.uio.no/griff/SIGMMrecords-1303.pdf>
24. Lightweight Directory Access Protocol (LDAP) (2017) <https://www.ietf.org/rfc/rfc2251.txt>. Accessed 12 April 2017
25. Lilholt PH, Jensen MH, Hejlesen OK (2015) Heuristic evaluation of a telehealth system from the Danish TeleCare north trial. *Int J Med Inform* 84(5):319–326
26. Mack RL, Nielsen J (1994) Usability inspection methods. Wiley & Sons, New York
27. Merakos L, Balaouras P, Tsimpanis K (2014) Opening up the University of Athens, the open and flexible higher education conference, Krakow, 23–24 October 2014
28. Munk TB, Mørk K (2002) Brugervenlighed på internettet: en introduktion. *Samfundslitteratur*
29. Nightson J (1994) How to conduct a heuristic evaluation. On the World Wide Web. <http://www.nngroup.com/articles/>. Accessed 12 April 2017
30. Nielsen J (1995) 10 Usability Heuristics for User Interface Design <https://www.nngroup.com/articles/ten-usability-heuristics/>. Accessed 12 April 2017
31. Open Archives Initiative (2017) <https://www.openarchives.org/pmh/>. Accessed 12 April 2017
32. Open Delos platform web portal (2017) <http://opendelos.org>. Accessed 12 April 2017
33. Operational Programme Life Long Learning (2017) Greek Ministry of Education. <http://www.edulll.gr/>. Accessed 12 April 2017
34. Panopto Video Content Management System (2017) <http://panopto.com>. Accessed 12 April 2017
35. Plupload (2017) <http://www.plupload.com>. Accessed 12 April 2017
36. RESTful Web Services (2017) <http://docs.oracle.com/javase/6/tutorial/doc/gijqy.html>. Accessed 12 April 2017
37. Ritchie H, Neyby TJ (1989) Classroom lecture/discussion vs. live televised instruction: a comparison of effects on student performance, attitude, and interaction. *Am J Dist Educ* 3(3):36–45
38. Sampson DG, Zervas P, Sotiriou S (2013) Sharing of open science education resources and educational practices in Europe. *Open Educational Resources: Innovation, Research and Practice*. 105
39. Section 508 Accessibility Guidelines (2017) <http://webaim.org/standards/508/checklist>. Accessed 12 April 2017
40. Shibboleth (2017) <https://shibboleth.net/>. Accessed 12 April 2017
41. Siemsen G (2013) Massive open online courses: innovation in education. *Open Educational Resources: Innovation, Research and Practice* 5:5–15
42. Sotiriou M, Sotiriou S, Doulamis N, Psaltis C, Georgopoulos A, Doxaras I (2008) Advanced Repositories for Science Teaching and Learning (the COSMOS Project): Main Operations and Technologies Used. Conference ICL2008, September 24–26, Villach, Austria
43. Spring Framework (2017) <http://projects.spring.io/spring-framework/>. Accessed 12 April 2017
44. T.A.W (2017) <http://www.tawdis.net>. Accessed 12 April 2017
45. The OpenCast platform (2017) <http://www.opencast.org>. Accessed 12 April 2017
46. The project Central Repository of Greek Open Courses (2017) <http://project.opencourses.gr>. Accessed 12 April 2017
47. University of Athens video services (2017) <http://delos.uoa.gr>. Accessed 12 April 2017
48. Virzi RA (1992) Refining the test phase of usability evaluation: how many subjects is enough? *Human Factors: the Journal of the Human Factors and Ergonomics Society* 34(4):457–468
49. WAVE (2017) <http://wave.webaim.org>. Accessed 12 April 2017
50. Web Content Accessibility Guidelines (WCAG) 2.0 - W3C (2017) <https://www.w3.org/TR/WCAG20/>. Accessed 12 April 2017



51. Wowza Streaming Engine (2017) <http://www.wowza.com>. Accessed 12 April 2017
52. eXistdb-The Open Source Native XML Database (2017) <http://exist-db.org>. Accessed 12 April 2017
53. Yousef AM, Chatti MA, Schroeder U (2014) Video-based learning: a critical analysis of the research published in 2003-2013 and future visions. The sixth international conference on mobile, hybrid, and on-line learning: eLmL 2014; Barcelona, Spain, from march 23, 2014 to march 27, 2014
54. Zhang D, Zhou L, Briggs RO, Nunamaker JF (2006) Instructional video in e-learning: assessing the impact of interactive video on learning effectiveness. *Inf Manage* 43(1):15–27



**Pantelis Balaouras** holds a PhD in Informatics and Telecommunications and a Diploma on Computer Engineering. He has expertise in designing and deploying multimedia-based e-learning and communication services, such as video conferencing, video streaming and on-demand services, multimedia digital collections, and IPTV services.



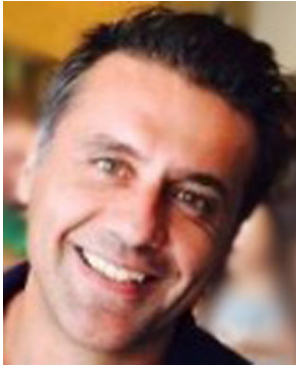
**Michael Gatzonis** is a Software Engineer and Laboratory Assistant Staff in Medical School at National and Kapodistrian University of Athens. His research interests include the use of technology for distance and online learning.



**Konstantinos Tsimpanis** holds an M.Sc in Informatics, an M.Sc in Economics and Telecommunication, and a B.Sc. in Mathematics. He has expertise in designing and deploying web-based internet and e-learning services, such as e-learning platforms, web application, e.t.c. He has also participated as a technical manager in a large number of projects related to e-learning services.



**Hippokratis Apostolidis** holds BSc degree in Mathematics and MSc degree in ICT in Education from the School of Informatics of Aristotle University of Thessaloniki. He is currently PhD candidate at the same department. His research interests include affective computing, computer supported collaborative learning and Human Computer Interaction.



**Thrasylvoulos Tsiatsos** is Assistant Professor at the School of Informatics of Aristotle University of Thessaloniki. His research interests include virtual learning environments, educational use of multimedia and internet technologies, Open and Distance Education with use of computer networks and real time protocols. He is member of the IEEE.