Enriching E-Learning Meta Data with User Generated Playlists

Maria Siebert
Hasso Plattner Institut Universität Potsdam
Potsdam, Germany
maria.siebert@hpi.uni-potsdam.de

Franka Moritz
Hasso Plattner Institut Universität Potsdam
Potsdam, Germany
franka.moritz@hpi.uni-potsdam.de

Frank Hambach
Hasso Plattner Institut Universität Potsdam
Potsdam, Germany
frank.hambach@hpi.uni-potsdam.de

Christoph Meinel
Hasso Plattner Institut Universität Potsdam
Potsdam, Germany
christoph.meinel@hpi.uni-potsdam.de

Abstract

Playlists are a concept commonly known from music players. Building on the concept of mix tapes they enable the user to select and group their favorite music and thereby creating new compilations of music. Web video portals have adopted this idea to let users create custom combinations of videos. This paper will focus on the task of adding playlist functionality in an E-Learning context. We describe how we implement the playlist solution as well as the use cases in which user of our portal can take advantage of playlists.

1 Introduction

Web 2.0 is one of the buzz word of modern web applications. In short it describes, that the user of a web page should be able to generate his own content and therewith enrich the information of the web page. Different types of user involvement are known. For example the user can participate in wikis and create collaborative knowledge. They can also write comments on existing content or rate the content of a web page.

For video archives, like the most common one YouTube[1], other types of participation are known. For example YouTube provides the possibility to rate videos and to create lists of your favourite videos [5]. This is useful, because YouTube is used as collection for all kind of videos. A few lectures can be found there, but many users of YouTube search for entertainment videos. So collections of the favourite videos occur natural.

For a tele-lecturing portal different types of content exist[4]. You can find recordings of lectures and conferences, so most users watch the videos, when they are interested in a specific topic or if they are doing preparation or wrap-up on their visited courses. This context lends itself to the implementation of web 2.0 technologies[9].

Because we have a different usage scenario than other video portals, playlists for video lectures have different use cases than playlists in other video portals. These use cases will be explained in section 3 more detailed.

The usage of playlists is not bound to the web. The concept of playlists is even older. Many portable music player provides such a function. It became popular, when audio compression like MP3 allowed to have large collections of music available on a single device, to manage the music data. So today playlists are available on nearly every music device.

So when implementing playlists in a portal different aspects has to be considered. At first we will have a look at the data, which a playlist can provide in section 4. Afterwards we will have a look, how the playlist functionality can be implemented without an impact on other functions of the portal but a good integration in the existing functions as well.

The creation of playlists does not provide simple playlists, it also provides extra meta data, which can be used in other functions. Therefore we will also have a look at the possibilities in section 6, the new meta data will create.

2 About the tele-TASK Portal

The basis of the tele-TASK portal (http://www.tele-task.de) is the archive of all the recorded lectures and events at our institute. Until today, over 3000 lectures can be found organized in about 400 series providing a huge collection of data. Many of them lectures are split in handy video clips called scenes. These videos are provided in different formats for better. With this separation of content is structure of different layers of data is generated.

The portal is implemented in Django, which is a python based web framework. It is implemented with a plug-in architecture, as described in [7]. This allows us, to separate the different functions and have modules for each task, like shown in Figure 1.

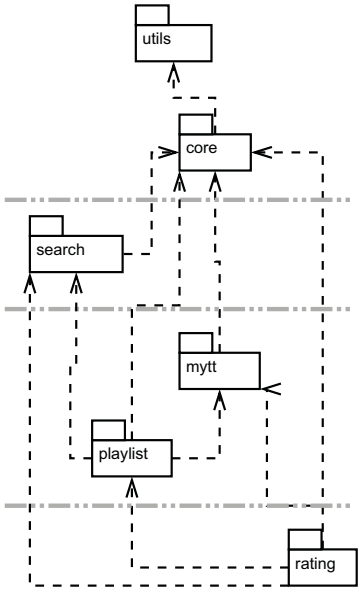

Figure 1. Extract of the structure using plug-ins

3 Use Cases

There are several use cases for lecture video playlists that can be suggested to the learners in a tele-teaching portal. Some of the options shall be explained in this chapter in order to show the utility of this feature for the users. But beforehand it is explained why it is necessary and useful to adapt features like group administration and a visibility parameter in several layers of hierarchy for playlists.

The first version of the playlists was implemented to work only for individual users. This means that a learner or teacher who creates a new playlist may decide whether his playlist is intended only for his private use or whether he would like to make it publicly available to all other users of the portal. In the future this feature shall be extended to be either used for individuals or for user groups.

Two examples why it makes sense to extend the functionality for user groups: First of all the flag public also exists for all layers of content items (series, lectures and scenes). Therefore content that is not allowed to be seen by the public cannot be seen in public playlists either. In the future the implementation of a user group structure throughout the portal is planned. Users (both, learners and lecturers) will be able to create user groups and invite colleagues, students or friends to that user group.

One field of application for that functionality may be that a lecturer creates a group where he invites all students of a current course and provides special content only for this group of people. Another example why user groups are useful are learning groups known from the university environment. Quite often students at university get together in learning groups to revise what they learned during the week, to work on exercises and homework, to prepare projects or talks together or to review the whole semester to prepare for exams. This concept is known as interpersonal learning [2].

It was shown that learning in a group is positive for supporting the individuals motivation and eagerness to engage into academic activities [8]. This gathering in groups shall be enabled for the digital world too in order to support successful learning methods for online learning as well.

Now all functionalities that are available for individuals have to be adopted to work for user groups as well. This is necessary to promote that students use the group structure that worked successfully for them in the real world in the digital world as well. When they do that rather than learning on their own, they have more functionality available. All playlist features therefore have to be adapted for the user group functionality. For this reason it makes sense to enable the same layers of visibility that will be available for the content items that can be included in the playlists (public to all, visible for all internal users, visible to specific user groups and visible to the logged in user only) for the playlists too.

With those basic functionalities, user group administration and layers of visibility, in mind, several use cases for video lecture playlists can be thought of:

3.1 Playlists for a Certain Topic

A student would like to learn everything about one specific topic, for example the Internet or the programming language Java. The first step would be to use the search. When searching for the one specific keyword, a large number of content items are found. The user now has the option to create a learning scedule by creating a playlist for the topic he aims to learn.

The playlist will help him to collect different content items required to get an in-depth insight into the subject. He may create several groups within the playlist to categorize all the content items into several sub-themes, for example basic concepts of Java, attributes, syntax, programming environment and compilers are possible sub-themes to the learning topic Java. This grouping helps sceduling and organizing the learning goal. The description to each group

may be used to further explain each sub-theme.

3.2 Exam Preparation

When learning for an exam students usually first revise all topics and then look in detail at themes they did not understand very well beforehand. A good option to prepare a learning schedule and collect all lectures or scenes that need to be revised in more detail are playlists. The groups in the playlist could be used for the subjects (called series in the tele-lecturing portal) that will be examined.

That student may not only use certain parts of the series he was following the whole semester, but also scenes from connected content, if more detailed insight into one topic is required. The description of the group can be used to note important aspects of the topic or write down important questions that the student would like to have answered before the exam.

3.3 Previous Knowledge Collection for Courses

Tutors may want to give their students a collection of material or lectures that are required as previous knowledge for the following lecture. This will enable the students to properly prepare for the next semester.

One example for that is the lecture on complexity theory. This course is usually open for a broad range of students with a different level of previous knowledge. Therefore it is recommendable to ensure that all participating students start into the semester with a similar level of knowledge by offering a playlist with some basic knowledge. The complexity theory playlist for example includes some mathematical basics, an introduction to turing machines as well as a collection of some problems that are tangent to complexity theory.

All those examples show that the utilization of playlists in connection with the group functionality may support the collaboration between the students and therefore enable a sense of social presence. It was shown that this sense of social presence positively influences the satisfaction of the students with the online learning process [3] which is a desired goal in tele-teaching.

4 Meta Data of Playlists

The idea of creating playlists is to enable the user to combine the existing content, to create new information sources. Therefore it is important to connect the available content with the new data structure. So we have two problems to be solved.

On the one hand we have two types of video contents in our project. The first one is the whole lecture, which is normally 90 minutes long and contains a lot of different informations. The second one is the video podcast, which are handy clips extracted from the lecture. Such a podcast is generally about 10 minutes long and contains mostly one specific topic. In the database both types of media are represented through one table, which contains the media data and can be distinguished by their connection to the specialized tables for lecture and podcast. Therefore we used the Django feature of generic foreign keys (see listing 1).

```
1 class Media(models.Model):
2   cT=models.ForeignKey(ContentType)
3   oId=models.PositiveIntegerField()
4   co=generic.GenericForeignKey("cT", "oId")
```

Listing 1. Generic Relation for Media

For generating playlists we want to enable the user to use both types of media. Therefore we enable the user to create playlists, which are capable to contain connections to these media entries instead of the specialized entries for lecture and segment.

On top of that, the structure for creating new playlist data should be as flexible as possible. For the user it should be possible to structure the playlists. This is useful to reflect the relations between the different objects in the list. There were two possibilities to create such groups:

1. First it is possible to use two data types, one for the playlist and one for the group. Then every playlist consists of one or more groups. This structure will create a fixed database structure.

2. It is also possible to have one data type which can contain itself as sub object. This creates a flexible structure like it is known from file and directories structures.

It is obvious that the on one hand the second approach is more flexible and would allow more possibilities. But more flexibility is not always helpful. It is harder to display all available information of such a data structure. The user has to understand the possibilities which are offered. For most users this can be to much freedom with to much administration tasks to do.

On the other hand the first approach gives a strict structure. But it allows to differentiate the data for playlist and groups. For example it is useful to allow a picture for a playlist but not for a subgroup. This approach is also more similar to the structure of courses and lectures. For a semester course we allow to group the lectures in subgroups. These groups are used to show sub structures in the lecture, which can be named individually by the lecturer.

We decided for the first solution. We therefore had to introduce three new models, one for the playlist, one for the groups and one for connecting the groups with the media objects (see listing 2).


```
class Playlist(models.Model):
  name = models.CharField()
  creator = models.ForeignKey(Profile)
  isPublic = models.BooleanField()
  logo = models.ImageField()
  created = models.DateTimeField()
  changed = models.DateTimeField()

class PlaylistGroup(models.Model):
  name = models.CharField()
  playlist = models.ForeignKey(Playlist)
  media = models.ManyToManyField(Media,
      through='PlaylistEntry')
  order = models.IntegerField(default=0)
  isPublic = models.BooleanField()

class PlaylistEntry(models.Model):
  playlistgroup = models.ForeignKey(
      PlaylistGroup)
  media = models.ForeignKey(Media)
  order = models.IntegerField(default=0)
```

Listing 2. Model for Playlists

5 Implementing a pluggable solution

A tele-teaching platform consists of different parts, like the video archive, the search function and different community functions. One of these parts are the playlists. To have a good separation of these parts, we used the basis plug-in architecture.

This architecture allows us, to separate the playlist functionality from other functions in the implementation, though this separation is not visible to the user of the portal. So the user of the portal could see the data and the design, which is implemented in different modules on a single page. Which data is visible depends on the installed modules and it is easily possible to add new modules or remove them without touching any other module.

The plug-in architecture is also used to enhance the search with new searchable meta data. So it allows us, to use the playlist meta data for search tasks.

The plug-in architecture also provided the possibility to provide own plug-ins for other functions. For example we provided a plug-in interface for rating functionality. This plug-in interface (see listing 3) provides the possibility to enhance the displayed data of a playlist with additional data. It is even capable to insert additional forms in the display page of a playlist.

```
class PlaylistInfo(TemplateContentProvider)
    :
  pluginTemplateName = "PlaylistInfo"

  def __init__(self, data, template = ''):
    TemplateContentProvider.__init__(self,
        data, template)
```

Listing 3. PlaylistInfo plug-in

The first improvement of the playlist is the possibility to rate, but further improvements using additional plug-ins are possible, like adding tags to playlists.

6 Usage of playlist data in the application

The data of the playlists can be used for different functions. Obviously playlists can be used for navigation and search functions, be we also use them, to have a flexible base for generating video DVDs.

6.1 Navigation using playlists

All user-generated public playlists are accessible through the archive sub pages of the portal. An overview page lists all available playlists. The playlist pages themselves present an overview of the lectures and podcasts they contain. For each such object a link leads to the individual video page where the lecture or podcast can be watched in an embedded player.

The plug-in architecture allows easy enrichment of those individual video pages with links to related content. The playlist module makes use of this possibility by listing all playlists that contain the lecture or podcast right next to the actual video. This opens up new navigation paths, especially when a large number of playlists is available.

6.2 Using playlists for search functionality

An important task for playlists is the enhancement of the search data. By creating own playlists user enhances the meta data of a media object with additional descriptions. If you use a lecture of the course Internet and WWW technology e.g. the lecture on TCP/IP, it can be found in a playlist about communication protocols as well. Through this playlist the lecture content is put into another context which gives more informations about the lecture itself.

So when searching for communication protocols it is likely that this playlist is found and the user finds the corresponding lecture.

To reach that goal, we used the plug-in architecture for the search function as well, as described in [6]. With the help of this architecture a module developer can provide sub

Figure 2. Example for embedded playlist function on the website

queries for enhancing the search query of the data base as well as provide new search types.

For the enhancement of the search function with playlist data, we used both of the described methods.

6.3 Using playlists for DVD creation

The video archive contains a vast amount of recorded lectures. These were only available through the web portal. It has been requested to be able to produce DVDs containing a custom set of lectures, enabling users to watch those videos offline. One DVD may, for example, contain all lecture videos of a conference. Such a DVD can be provided to the participants and other interested parties to complement the proceedings of the conference. Other DVDs may function as advertisement to demonstrate the possibilities of our system.

As this is a recurring request we decided to implement an automated DVD generation function as part of the web portal available only to staff members. To select and group the videos that are to be included in a DVD we decided to utilize playlists. That way we could avoid having to implement a whole new interface for composing the DVD contents.

The DVD generator, as part of the web portal, is implemented in Python and Django. Given a playlist the DVD generator is able to produce a small set of webpages for navigating and displaying the selected videos. This set consists of an overview page and individual pages for each lecture video.

These pages are independent from the rest of the web portal. They do not contain any of the broader navigation features that would take a user away from the selected set. Because they contain much less elements than regular video pages on the web portal they can feature a much lighter layout and thus look very different from the rest of the web portal. The lecture video pages simply embed a Real player to play the specific video and a link to the overview page.

Because these pages have to be served statically from a DVD, the DVD generator basically downloads the dynamically generated pages onto the web server itself. During this process it ensures that links within these pages are rewritten to relative links to work in the DVD environment. All downloaded pages are packed into a zip archive on the web server. This archive can then be downloaded by users of the DVD generator. It does not contain the videos itself. Instead it contains a text file listing the videos that have to be included. As the staff members have access to the file system of the video archive they can easily add the files themselves. Because video files tend to be very large this should be much faster than having the DVD generator pack these files into the zip archive himself and letting the user download the large archive.

Being able to play the videos from DVD restricts us to lectures and podcasts where the video is available as Real video. Lectures and podcasts that can only be played within a Flash player cannot be included in the statically served pages of a DVD. Because we just utilize the existing playlist functionality a restriction on a specific video type can not be applied to a playlist when it is created. The DVD generation function, being a feature that is used by staff members only, does not warrant the inclusion of such an extra functionality to playlists. Ultimately users have to be aware of this restriction when selecting videos for a DVD.

7 Conclusions and Future Work

In this paper we showed, how playlists for video lecturing can be used and what has to be considered when implementing them.

One of the biggest problems we have to solve in the future is the low rate of user activity in our portal. Only few students use the existing functions to generate own data. For

playlists this means, that there are only a small number of playlists, which are mostly generated by co workers of our institute, but not by students.

So a big task will be to reach the students for using the portal as part of their learning tasks. Therefore we see three main approaches, will will be described in the following parts. Having these three tasks solved, the usage behaviour of the users in our portal can be analysed to get a better understanding of the usage of playlists in this special context.

7.1 Evaluating the usability of the portal

When designing user interfaces, it is an important task to look at the usability of the interface. A good usability of the playlist function will help the user in creating their own playlists. It will also encourage to use the function more often because of the minimized effort. That is why we are planning the execution user tests for the playlist function of the portal.

The first beta tests have already taken place and showed us, that there is a lot to do. To get an intuitive interface for the users, who wants to create their own playlists or use the playlist of their fellow students, we will have to do more research on usability of different interfaces and interaction possibilities.

7.2 Creation of incentives for using the function

Many web 2.0 applications give incentives to their user for generating more user activity. The easiest type is the usage of ranks, like it is done in different forums. A user gets a special rank, when he writes a specific number of messages in the forum. Other incentives like special prices or more rights are known as well.

On the opposite, user incentives can also produce useless data. Many forums have problems with user spamming the forum threads with short texts to get a higher rank. So it can happen that incentives will lower the quality of data. This could result in bigger damage than the additional collaboration is worth of.

It has to be evaluated, which incentives will have a good effect on user activity without leading to the production of dummy data.

7.3 Group functionality

The actual version of playlists allows the creation of private and public playlists. But this is not enough. It is a normal use case, to create playlists only for a special group of students, but not for everyone. A learning group for example will have their own playlists, which do not have to be visible to every user of the portal.

References

[1] Meeyoung Cha, Haewoon Kwak, Pablo Rodriguez, Yong-yeol Ahn, and Sue Moon. I Tube, You Tube, Everybody Tubes: Analyzing the World's Largest User Generated Content Video System. In *ICM*, pages 1–13, San Diego, California, USA, 2007.

[2] David Gallula and Ariel J. Frank. Enriching the E-learning Experience in the Framework of Web 2.0 Using Usability 2.0. *2009 Fourth International Multi-Conference on Computing in the Global Information Technology*, pages 229–234, August 2009.

[3] Carol Hostetter and Monique Busch. Measuring up Online: The Relationship between Social Presence and Student Learning Satisfaction. *Journal of Scholarship of Teaching and Learning*, 6(2):1–12, 2006.

[4] Hermann Körndle, Ulrich Marder, and Günther Robbert. Entwicklung und Einsatz einer Videodatenbank im WWW - ein Erfahrungsbericht. In *Informatik '98 Workshop "Multimedia-Systeme"*, volume 31, pages 109–124, Magdeburg, January 1998. Gesellschaft für Informatik.

[5] Michael Miller. *Youtube 4 you*. Que Corp., Indianapolis, IN, USA, 2007.

[6] Maria Siebert and Christoph Meinel. Realization of an Expandable Search Function for an E-Learning Web Portal. In *2010 IEEE/ACIS ICIS/IWEA 4th International Workshop on E-Activity*, Yamagata, Japan, 2010. IEEE Computer Society.

[7] Maria Siebert, Franka Moritz, and Christoph Meinel. Establishing an Expandable Architecture for a Tele-Teaching Portal. In *2010 Ninth IEEE/ACIS International Conference on Computer and Information Science Article*, Yamagata, Japan, 2010. IEEE Computer Society.

[8] Hyo-Jeong So and Thomas A. Brush. Student perceptions of collaborative learning, social presence and satisfaction in a blended learning environment: Relationships and critical factors. *Computers & Education*, 51(1), 2008.

[9] Carsten Ullrich, Kerstin Borau, Heng Luo, Xiaohong Tan, Liping Shen, and Ruimin Shen. Why web 2.0 is good for learning and for research. *Proceeding of the 17th international conference on World Wide Web - WWW '08*, pages 705–714, 2008.

