A method for a small web site to add some video sharing features

Juan Lucas Madurga Martín-Serrano

2008-01-31
A method for a small web site to add some video sharing features

Examensarbete utfört i datavetenskap
vid Tekniska Högskolan vid
Linköpings universitet

Juan Lucas Madurga Martín-Serrano

Examinator Bengt Lennartsson

Norrköping 2008-01-31
Upphovsrätt

Detta dokument hålls tillgängligt på Internet – eller dess framtida ersättare – under en längre tid från publiceringsdatum under förutsättning att inga extraordinära omständigheter uppstår.

Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för ickekommersiell forskning och för undervisning. Överföring av upphovsrätten vid en senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmannens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns det lösningar av teknisk och administrativ art.

Upphovsmannens ideella rätt innefattar rätt att bli nämnd som upphovsman i den omfattning som god sed kräver vid användning av dokumentet på ovan beskrivna sätt samt skydd mot att dokumentet ändras eller presenteras i sådan form eller i sådant sammanhang som är kränkande för upphovsmannens litterära eller konstnärliga anseende eller egenart.

För ytterligare information om Linköping University Electronic Press se förlagets hemsida http://www.ep.liu.se/

Copyright

The publishers will keep this document online on the Internet - or its possible replacement - for a considerable time from the date of publication barring exceptional circumstances.

The online availability of the document implies a permanent permission for anyone to read, to download, to print out single copies for your own use and to use it unchanged for any non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional on the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility.

According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement.

For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its WWW home page: http://www.ep.liu.se/

© Juan Lucas Madurga Martín-Serrano
Summary/Abstract:

The aim of this diploma work is to compare and evaluate different methods to enable small web sites to make available videos which will be shared the users. Storage and bandwidth problems of small web sites are taken into account. The requirements of the system were defined as: configurable, feasible, easy use and easy to integrate. Existing solutions and different implementation alternatives are analyzed. With a tool based upon a CMS, using recordings residing on powerful dedicated video providers and with AJAX “technology”, the criteria are fulfilled. As part of the investigation, a prototype tool based upon TYPO3 has been implemented.
# Content

Summary/Abstract: .......................................................................................................................... 1

Content .................................................................................................................................................. 2

1 Introduction ......................................................................................................................................... 2
   1.1 Background: ............................................................................................................................... 3
   1.2 Purpose of the thesis: .................................................................................................................. 3

2 State of Art ......................................................................................................................................... 4

3 Definition of problem scenario. Study of alternatives ........................................................................... 6
   3.1 Methodology .................................................................................................................................. 6
   3.2 Scenario ......................................................................................................................................... 6
   3.3 System characteristics .................................................................................................................... 6
      3.3.1 Feasible .................................................................................................................................. 6
      3.3.2 Easy integration ....................................................................................................................... 6
      3.3.3 Easy configuration ................................................................................................................... 7
      3.3.4 Easy use ................................................................................................................................ 7
   3.4 Alternatives and decisions ............................................................................................................ 7
      3.4.1 Feasible .................................................................................................................................. 7
      3.4.2 Easy Integration ...................................................................................................................... 8
      3.4.3 Easy configuration ................................................................................................................... 11
      3.4.4 Easy use ................................................................................................................................ 11
      3.4.5 Existing solutions analysis .................................................................................................... 13

4 System development life cycle ............................................................................................................ 14
   4.1 Methodology .................................................................................................................................. 14
   4.2 Requirements ................................................................................................................................. 14
      4.2.1 Functional requirements ......................................................................................................... 14
      4.2.2 Non functional requirements .................................................................................................. 15
   4.3 Use cases diagram .......................................................................................................................... 16
   4.4 Use case description ....................................................................................................................... 17
   4.5 Construction ................................................................................................................................ 21
      4.5.1 High-Level Design .................................................................................................................. 21
         4.5.1.1 Conceptual model .......................................................................................................... 21
         4.5.1.2 System sequence diagram ............................................................................................... 22
         4.5.1.3 Contracts ........................................................................................................................... 23
      4.5.2 Low-Level Design ................................................................................................................... 25
         4.5.2.1 Architecture: Model-View-Controller: ............................................................................. 25
         4.5.2.2 Interactions diagrams ........................................................................................................ 26
         4.5.2.3 Classes Diagram ............................................................................................................... 32
         4.5.2.4 ER Diagram ....................................................................................................................... 33
      4.5.3 Implementation ......................................................................................................................... 34
   4.6 Demonstration ................................................................................................................................ 37

5 Conclusions .......................................................................................................................................... 49

References .............................................................................................................................................. 50
1 Introduction

1.1 Background:

An already built small web site allows exchange of information between users groups with something in common. These users share interests, ideas, knowledge or simply live in the same geographic zone. The administrators want to promote the web site providing new complementary services. A good alternative may be sharing videos. Size of video files may cause storage and bandwidth problems. The small web site does not have a big amount of resources to solve them. Having a big group of developers is not possible; therefore any changes made in the current site should be easy to integrate and not problematic. Which may be an appropriate method for a small web site to add video sharing features in an easy and configurable way?

1.2 Purpose of the thesis:

The main purpose of this thesis is to compare and investigate different methods to add some features to an already built small web site that enables users to share videos. The evaluation criteria are: easy integration, configurable, easy use and feasible according to few resources of small sites. As a starting point we will search for existing solutions related to video sharing communities. Once we define a more complete scenario and some system characteristics according to small web sites limitations, we will analyze alternatives and make some decisions to satisfy obtained requirements. At this point it will be possible to evaluate existing solutions previously searched. Furthermore, we would develop a system which reflects earlier specifications. The obtained system life cycle will be explained; how the developed results match the system requirements and the experience gained will also be discussed.
2 State of Art

Currently, there is a great amount of solutions related to video sharing. However, we should bear in mind our particular limitations so we can discard some of them. To provide a better vision of the investigated solutions we classify them according to how they are integrated.

A way to create and manage a web site is by a Content Management System, CMS, i.e. a system used to manage the content of a Web Site [1]. There are basic web contents such as paragraphs, images or lists but there are also more complicated ones such as articles or calendars. The latter are normally available as add-ons and are easy to integrate in CMS and use in sites, especially for not web experts. They are usually called plugins or extensions. Several solutions to our issue can be found in some CMS Open Source platforms. As the number of Open Source CMS is very large, the thesis has taken into consideration the most popular ones. Some solutions like “Ach Tube” for Joomla*, “Open Package Video” for Drupal or “Plumi” for Plone try to afford all the features a big video host has: uploading videos, convert them into FLV format, statistics… There are also other simpler extensions used to add embedded videos which provide the video link: “Seyret Video Component” for Joomla!, “Googtube” for Drupal or “TIMTAB embedded video” for TYPO3. Others significant plugins are the video searchers, “Google Custom Search” for Joomla! or “Google Ajax Search Module” for Drupal.

Another different technique to make changes in a web site and provide the wanted functionality is installing some scripts not managed with a CMS. We get some scripts and try to integrate them in our current web code. Options abound but most of them are commercial. There are various YouTube clones like “Video Share Enterprise”, “Video Uploader Like Youtube”, or “ClipShare”. Besides, there are big solutions trying to give the same features as big video host, buying the license cost several hundred dollars. We find other as deep but very attractive solutions. These handle video feeds, but do not storage video files: “RSS Media Grabber” or “Youtube Keyword Script”. These scripts only use most common hosting video site feeds such as “Last added”, “Featured”, “Most popular”...

Many Internet companies find complex or not profitable to manage a big amount of media files, so they delegate this task to someone else. The idea is outsourcing the content delivery to a third party. This is a business model called “content delivery network” [2]. Companies dedicated to this scope, such as Akamai, work for large Internet companies and provide the content through transparent mirrors content stored in customer’s servers. More specific providers dedicated to video

---

1 http://extensions.joomla.org/component/option,com_mtree/task,viewlink/link_id,2509/Itemid,35/
2 A CMS: http://www.joomla.org/
3 http://drupal.org/project/op_video
4 A CMS: http://drupal.org/
5 http://www.plumi.org/
6 A CMS: http://plone.org/
7 Flash Video format http://es.wikipedia.org/wiki/FLV
8 http://extensions.joomla.org/component/option,com_mtree/task,viewlink/link_id,1628/Itemid,35/
9 http://drupal.org/project/googtube
10 http://typo3.org/extensions/repository/view/timtab_embeddedvideo/2.0.4/
11 A CMS: http://typo3.org/
12 http://drupal.org/project/googleajaxsearch
13 http://www.alstrasoft.com/videoshare.htm
14 http://www.customizedscripts.biz/uploadvideo_script.html
15 http://www.clip-share.com/
16 http://phppod.com/RSS-Media-Grabber.html
17 http://www.phpgator.com/superkeyword.html
18 http://www.akamai.com/
are companies with video hosting websites. They offer their platform and technologies allowing clients to create the video sharing services. These are called “White Label Providers” [3]. Examples of this type of business are Stashspace\(^1\), Kewego\(^2\), VideoEgg\(^3\) and vMix\(^4\).

\(^1\) Before known as HomeMovie.com \url{http://www.stashspace.com/}
\(^2\) \url{http://www.kewego.com/}
\(^3\) \url{http://www.videoegg.com/platform/}
\(^4\) \url{http://www.vmix.com/}
3 Definition of problem scenario. Study of alternatives

3.1 Methodology

One of the first purposes of the project was to identify the problem scenario and defining the main characteristics that our system should follow. After studying different alternatives, some decisions will be taken.

3.2 Scenario

An already built web provides some functionality to its users. The web has several groups containing users who share interests, location, research or discussion groups... People who are in charge decide to introduce some changes in order to provide more services and promote their site. They have been thinking about the possibility of allowing users to share videos. Although sharing videos will not be the main purpose, it will work as a complement to other purpose. Competing with big sites like YouTube is not the goal.

Since this web site is small, having a big group of developers is not possible. Only few people can update the contents. As the site suffers from a lack of resources, we assume they cannot afford a big bandwidth and they may have some difficulties with video storage.

Administrators want to integrate new features as easily as possible without making changes or creating any problems to the actual content. As each group of users may have different necessities, the system may need some configurable parameters.

The flow that the system has to provide begins when a user wants to share a video. The user adds a video and selects which groups he wants it to be shared in. Once added, the video could be seen by any of the users who belong to the groups where the video has been added to; users are informed about which videos are the last ones that have been added. The video will have some characteristic attributes to provide users information such as duration, stats, author... The lack of administrators could be partially solved by giving more responsibility to the users. Therefore, if a user considers a video to be inappropriate to the group he could vote to delete it. The video will only be deleted if a majority of users classify it as inappropriate. The system should permit the administrators to manage the videos and control all the process of adding videos and deleting videos.

3.3 System characteristics

As a result of analyzing the scenario, the following characteristics have been developed.

3.3.1 Feasible

The system should be feasible following some restrictions of a small site: not big bandwidth, no mass storage, no commercial use. Consequently, how will video management be done?

3.3.2 Easy integration

As we have an already built web site this new functionality should be easy to integrate without
causing traumatic changes which may affect users. What is needed to provide functionality avoid many changes in the site?

### 3.3.3 Easy configuration

Anticipating possible changes in the way of use of these new functionalities, the system should be easy to be configured so that an inexperienced administrator could configure some aspects without changing the code. How could the administrator configure the system in an easy way?

### 3.3.4 Easy use

Finding, adding and watching videos should be easy, especially for users. Moreover, this needs to be independent from the main purpose of the site. The flow process should not be slow or cumbersome. Which are the technologies that allow us to make an intuitive interface?

### 3.4 Alternatives and decisions

Once given several characteristics of our system, some alternatives and decisions are analyzed and taken. Related work and existing applications are being analyzed too.

#### 3.4.1 Feasible

As defined in the previous step, the system should follow some requirements to satisfy some restrictions assumed for a small site. Video files storage may be a problem, but nowadays is not so expensive to buy a mass storage device, 1 TB HD for 300€\(^1\). On the other hand, other mentioned restriction was not being able to deliver content with a big bandwidth. There are lots of video compression formats that can effectively reduce the bandwidth requirements to transmit video. One of the most common ways is encoding the video files into FLV format. This format is very popular and is used by many websites like YouTube, Google Videos, etc... FLV uses some compression codecs such as On2 VP6 and Sorenson Spark \(^4\). For example Sorenson codec has 2.5:1 compression rate \(^5\). In order to supply an acceptable service, although not achieving best quality videos, is required at least 329 Kbps \(^5\). Bandwidth will cause problems when the number of users increases.

Why not sharing videos from the main video hosts? Main video sharing big web sites, like YouTube, give the possibility to use their videos embed like a web widget\(^2\); on the same way, videos can be used in our site as embed content. Now the bandwidth problem is solved. Nevertheless it is necessary to read YouTube terms of use (or any other similar provider) which must be carried out. YouTube prohibits using their embed videos for commercial aspects \(^7\). As we explain in the scenario, one web site characteristic is that its aim in not commercial use. Moreover, supplying videos is not even a primary purpose.

It is time to explain how to retrieve videos from video hosts. Most of video hosts have ways

---

1. Hitachi Deskstar 7k1000 costs 374$ for 1 TB
   [http://www.hitachigst.com/portal/site/en/menuitem.8027a91c954924ae4bda9f30eac4f0a0/The%20web](http://www.hitachigst.com/portal/site/en/menuitem.8027a91c954924ae4bda9f30eac4f0a0/The%20web) [viewed 08/01/2008]

2. The web widget is a portable chunk of code that can be installed and executed within any separate HTML-based web page by an end user without requiring additional compilation. [http://en.wikipedia.org/wiki/Web_widget](http://en.wikipedia.org/wiki/Web_widget)
of syndication\(^1\) through web feeds to provide other people with a summary of recently added videos. Apart from giving the possibility of getting last updated videos or videos with specific characteristics (like most viewed), they also offer the option of making searches on their videos. Sites like YouTube offer even more features than those mentioned above [8]. These services provide so many features that are considered as an API\(^2\). This API offers the developers different data such as videos, user’s favourites, video responses, video comments, playlists, subscriptions, user's profile... If we value the possibility of using all these features we notice that is not necessary. It is not valuable for our purpose and it is more significant to offer videos not only from one site. Users will not find the site attractive if only YouTube videos are available, they will prefer to use YouTube web site instead.

Web feed is a document based on XML\(^3\). These documents can be structured in several ways; the main formats are RSS\(^4\) and Atom\(^5\). The appearance of Atom was motivated by dissatisfaction with RSS [9]. Some modules have been added to the RSS format to provide more robust syndication. There is one RSS module called “Media RSS”\(^6\) Yahoo to provide a better support for media feeds. To allow correct interpretation of this added namespace, an Xpointer\(^7\) is defined at the beginning of the feed, xmlns:media="http://search.yahoo.com/mrss/". In this way, feed parsers\(^8\) could know the structure of new specifications. Another popular add-on to the RSS specification in order to increase the functionality with searches, is “OpenSearch”\(^9\) developed by Amazon.

With this background of syndication technologies, searching how to retrieve video feeds of each video hosts may be a hard work. On the other hand, focusing only in some video hosts may decrease interest in using the system. Google is always offering new functionalities. One of them is “Google Custom Search” [10]. We can retrieve results from a specification of the Google search engine that prioritizes or restricts search results. Defining all the video hosts found on the web, we could get a Video search engine. Not all are advantages as results are presented in a Google way, including Ads. The only way we can get the results in a feed format, is in a pay version\(^10\) for companies. Nevertheless, Google Videos has normal feeds and makes searches not only for their videos but also for other providers. If we know how the Video Google feeds work to search and what type of data they retrieve, we could have a video searcher which looks for videos of the entire web.

### 3.4.2 Easy Integration

Adding new functionalities to a web site or a web application, involves the same problems as other software. We will need experts to integrate the new code with the old one. Sometimes people who are managing a web site are not software experts and may find difficulties. They may need support from experts, with the consequent of cost increase. However, if we have our web site in a CMS we will find adding new contents easier.

---

1. Is the process for automated updates, another way of getting the information you want [6]
6. [http://search.yahoo.com/mrss](http://search.yahoo.com/mrss)
7. [http://www.w3.org/TR/xptr-xmlns/](http://www.w3.org/TR/xptr-xmlns/)
8. The feed parser selected for the implementation is “Simple pie” [http://simplepie.org/](http://simplepie.org/)
9. [http://www.opensearch.org/Home](http://www.opensearch.org/Home)
Using a web CMS has lots of advantages [11] [12] some of them related to the integration:

- **Non technical people required:** easier to integrate new features through extensions or plugins.
- **Standardization:** CMS software makes standardization easier; designs, text sizes, fonts, formats, and layouts can be created consistently throughout the site.
- **Separation of Content, Structure and Design:** redesign a web or the structure is not so traumatic because there is a separation.

Developing a video sharing extension for a CMS will provide this CMS the possibility of adding these features in a simple and configurable way.

At this point we need to select the CMS. There is a large amount of these tools, approximately 1000 worldwide¹. It is logical to choose an Open Source because of small site characteristics. But it is not easy to choose a CMS between all the existing offers. There are some considerations to be taken in the election [13] and some mistakes better to avoid [14]. CMS provides a wide range of features, so it's important to choose the one which better fits our requirements. Therefore, the most important activity is to identify our business goals and requirements. This task is not easy; there are many options with highly variable capabilities in a rapidly evolving environment. Some companies help others to make their selections with reports and researches².

This thesis does not really have requirements to choose between all the CMSs, in any case only the CMS framework to develop the extension. However, it is interesting to make a comparison of the most popular Open Source ones before choosing any.

When looking for the best CMS on the web, we find many different opinions. Some of them are always mentioned, but each user has their own opinion. Searching for awards given to Open Source CMS, we find the ones given by PACKT³. In the last years, Drupal and Joomla! have won the main categories [15]. However, they are not very coherent since the best CMS are not the best PHP CMS. Nevertheless, we can take these two CMS as options. Other CMS described by users as powerful as the previous ones is TYPO3.

Trying to select one from these three, we can make some comparisons. Using the comparison of CMS Matrix⁴ web page, we found they use similar technology, PHP⁵ + MySQL⁶ + Apache⁷. TYPO3 seems to be the most powerful, as it has almost all the features the other two have. Considering security, Joomla! is not as complete as the others. In addition to this, we can see demos⁸ to get an overview of using one CMS or the other. Besides, on other interesting point is observing which one is the most popular nowadays and the evolution of this popularity. A really interesting Google tool is the Google Trends⁹ which provides searches and news results of several concepts using graphs. Results of the three CMSs are analyzed¹⁰ (Figure 3.1). We can state that nowadays¹¹ the most popular in searches is Joomla!, being the difference between Drupal and TYPO3 almost irrelevant. Joomla! appeared in 2005 and gained much popularity very fast. TYPO3 and Drupal appeared some years ago. Still, the news volume is more equalized, being TYPO3 news bigger in the last months.

⁹ [http://www.google.com/trends](http://www.google.com/trends)
¹⁰ Viewed in November 2007 [http://www.google.com/trends?q=joomla%2C+drupal%2C+typo3&ctab=0&geo=all&date=all&sort=0](http://www.google.com/trends?q=joomla%2C+drupal%2C+typo3&ctab=0&geo=all&date=all&sort=0)
¹¹ January 2008
Comparison results confirm with the opinions of CMS users on the web. Joomla! may be more popular because of its easy use; however, it has fewer features than the others. In this thesis we do not have as many specific requirements as a company or an organization may have to analyze the different options for the selection. Nevertheless, one important aspect for the development of the prototype of this thesis, is the framework the CMS provides.

A typical architecture for developing Web applications is the MVC, an architectural pattern to separate the data model, business and user interface. CMS philosophy fits really well into this pattern, due to the content separation. Therefore, the CMS framework would be really useful for a MVC development. Joomla! 1.5 framework [16] provides some classes to follow this architecture. Researching Drupal framework we can find the following criticism: some perceive the framework as not OOP [17]. In spite of this, theoretically it is possible to develop following MVC pattern [18]. In fact, TYPO3 is now developing a new framework to develop following MVC pattern [19]. There are other not so simple ways TYPO3 uses to do MVC development, as it is done in the “iCalendar” extension. Other extensions which provide functionality to program with this architecture, appeared previously [20] [21], although they were not very popular. It may be interesting to see how the new TYPO3 MVC oriented framework works.

In conclusion, this thesis chooses to use CMS to give an easy and configurable integration of video sharing features to the small web. Even though it is quite complicated to select a CMS because our lack of specific requirements and due to the small differences between the main characteristics, we will select TYPO3 for our prototype development. This selection is based on its

---

1 Model View Controller. More explanations in the System Life Cycle stage.
2 Object Oriented Programming
3 This extension is developed following MVC architecture but using the old framework http://www.wischnat.de/1576.0.html
4 These extensions/libraries were not created by a coordinate team. These extensions provides classes to implement MVC paradigm in other extensions but using the old official framework [20] [21]
powerful opportunities despite having a long learning curve for new developers.

### 3.4.3 Easy configuration

When developing software, it is always very usefully if it is configurable. This way, you can get more audience. As we mentioned above, the system should be easily configurable. This fact is analyzed and resolved in the previous point. With a CMS we allow not expert users to configure the added plugins. The CMS choice was TYPO3 but, does TYPO3 provide developers an easy way to supply easy configuration in not expert users?

In TYPO3 there is a declarative language to configure the template engine as well as extensions [23]. TypoScript could be compared to the standardized template technology XSLT/XML. Learning a new language is always a problem especially for not expert people. Another option to configure extensions is using flexforms [25]. This methodology allows us to declare some customizable variables through a XML file. When the users add a new plugin they will find in the TYPO3 BE (back-end) some intuitive forms to select the configurable values (Figure 3.2). This is an easy way to configure the plugins for not expert TYPO3 users.

![Figure 3.2: Plugin BE configuration form example](image)

### 3.4.4 Easy use

Characteristics about the ease of use have been defined previously. The system has to be introduced in the current site trying not to affect the previous content. These requirements are solved with AJAX “technology” a key component of Web 2.0 applications [27]. Getting asynchronously data from the server and refreshing only the interesting part of it, we get this desktop application effect.

There are plenty of AJAX frameworks. When looking for an appropriate framework for our necessities it is important to know that AJAX should be used only to get just series of data. The rest of data properties, such as order, layout or size should be left entirely to best hands, graphic designers and usability experts [28]. Researching on the web we found some JavaScript libraries features lists of comparisons. The main conclusion of these comparisons is that the most popular ones are: DoJo, Prototype, YUI, jQuery. DoJo seems to be the one with more Widgets and with

---

1. TYPO3 tutorials for beginners warn of the complexity [22]
2. XSLT is a stylesheet to transform XML into something else, e.g. HTML or PDF. TypoScript declares PHP/TYPO3 configuration [24]
3. Asynchronous JavaScript and XML. [http://www.w3schools.com/ajax/default.asp](http://www.w3schools.com/ajax/default.asp)
4. AJAX is not a technology, is considered as a group of technologies [26]
5. A list of frameworks organized in groups is available in [http://ajaxpatterns.org/Ajax_Frameworks](http://ajaxpatterns.org/Ajax_Frameworks)
6. Some aspects are subjective or change in the time. Therefore, it is better to analyze several comparisons [29] [30][31]
more support to make visual effects like tabs, tree views, etc... However, the others seem to be easier to use. A new format for structured data transmission and alternative to XML is JSON\(^4\). JSON support is a weak point of Prototype. Besides the mentioned libraries, there is a new one called Ext JS\(^5\) which extends YUI, jQuery and Prototype but can run stand-alone. We can use Google trends to have an overview of each library appearance and popularity\(^6\) (Figure 3.3). Dojo seems to be the most popular among them. Nevertheless, new ones, like jQuery, have grown up very fast and Ext JS has just born.

![Trend history](http://www.google.com/trends?q=jquery%2Cextjs%2CYUI%2Cdojo&ctab=0&geo=all&date=all&sort=0 )

**Figure 3.3:** jQuery, Ext JS, Yui and DoJo Google trend history graph

We chose to develop the system in TYPO3 so now it’s time to research how these libraries are used in TYPO3. The JavaScript’s libraries in TYPO3 are extensions that once installed, can be included and used in our extensions\(^7\). One new way of including JavaScript libraries in TYPO3 being developed now, is called JSManager\(^8\). The reason is to provide a general way of including all required JavaScript to a page, regardless how many different extensions there are. Continuing our research we encounter in TYPO3’s repository these AJAX libraries: Dojo, Prototype, script.aculo.es\(^9\) jQuery, Ext Js and xajax\(^10\). The Extension Coordinator Team\(^11\). ECT, has an extension\(^12\) to give dependencies to the most important extensions to be used; this is another aspect which will help us to take the decision about the library selection. This ECT recommends AJAX, Ext JS and jQuery. Ext JS may be the most interesting library to use because it extends other ones

\(^{1}\) [http://www.prototypejs.org/](http://www.prototypejs.org/)
\(^{3}\) [http://jquery.com/](http://jquery.com/)
\(^{4}\) JavaScript Object Notation. [http://json.org/](http://json.org/)
\(^{5}\) [http://extjs.com/](http://extjs.com/)
\(^{6}\) Viewed November 2007
\(^{7}\) [http://www.google.com/trends?q=jquery%2Cextjs%2C YUI%2Cdojo&ctab=0&geo=all&date=all&sort=0 ](http://www.google.com/trends?q=jquery%2Cextjs%2CYUI%2Cdojo&ctab=0&geo=all&date=all&sort=0 )
\(^{8}\) The documentation of how to use this extension and other use aspects are in its manual [32]
\(^{9}\) This extension has begun its development during this thesis. It is in a very early development stage. It is difficult no extract a good conclusion of its performance. The project code is in sourceforge:
\(^{10}\) [http://script.aculo.us/](http://script.aculo.us/)
\(^{11}\) Roadmap, projects, members and other interesting information is available in [http://wiki.typo3.org/index.php/ECT](http://wiki.typo3.org/index.php/ECT)
\(^{12}\) It is called “ECT coordinated extensions”. This extension orientates in the jungle of the many libraries within the repository. The extension and the manual in [http://typo3.org/extensions/repository/view/ect/0.1.1/](http://typo3.org/extensions/repository/view/ect/0.1.1/)
and because of its features and easy use. Moreover, Ext JS provides layouts and CSS\(^1\) which may help us to make the graphic design, although this work is better to be left to a designer expert.

### 3.4.5 Existing solutions analysis

The thesis has defined which alternatives are better for our issue. Therefore, at this point we are now prepared to analyze the existing solutions searched in a previous stage.

White Label Provider possibilities are good for big companies which want to give an optimal service. Outsourcing the service will be a good method if the main purpose deals all the time or if incorporating the new service quickly is very important. However, a big amount of money will be needed. As our scenario is for small web sites with lack of resources and as sharing videos is complementary to other purpose, we can discard this expensive option.

Adding features with scripts may be useful for simple requirements. Adding third party scripts usually is not an appropriate method for a professional site. It is probable that integration problems appear. Programmers know the difficulties involved when other person code needs changes. In any case, some of the solutions, such as “Video Uploader Like Youtube” or “Video Share Enterprise” might have a good integration performance, but to study it deeply we need to spend hundreds of dollars to get a license and some web expert developers. In summary, we can qualify these solutions as appropriate to make a new web site, but not to provide new features to an already built one.

One of the paths taken in this thesis is to use a CMS. Analyzing the related extensions, we found existing good quality plugins such as “Ach Tube” for Joomla!. However, it is for commercial use. Nevertheless, none of them fulfils the scenario requirements. Most of them manage video files and some of them use the video feed concept but in a manual way. Examples of this are “Googtube” for Drupal or “Seyret” for Joomla!. Although there are some video searchers extensions such as “Google Custom Search” for Joomla! or “Google Ajax Search Module” for Drupal, they do not allow adding and sharing videos, which is one of the most important features the system should supply.

\(^1\) Cascading Style Sheets [http://www.w3.org/Style/CSS/]
4 System development life cycle

4.1 Methodology
As a result of the acquired knowledge on the previous steps, this thesis project is developing a system which satisfies some functional and non-functional requirements. In order to carry out the system development life cycle we will follow Larman’s UML Process [33].

4.2 Requirements

4.2.1 Functional requirements
Several groups and registered users, who belong at least to one of these groups, form the web site community. The users participate in their groups with opinions, ideas, discussions… if a user finds interesting for a group to share a video; they should be able to use the searcher to find the video. The searcher should use Google Video searcher engine. While the user is searching, playing or viewing the host page, have to be available. When the user finds a video interesting, the system should allow the user to add the video to one of their groups. These added videos can belong to different groups, but only once in each group and to only one group’s user.

Users should be able to watch the videos added by other group users. A searcher of added videos should be available for all users to look for added videos in each group they belong to. This searcher should allow users to filter videos with a certain text in the title or in the description. The videos which the searcher provides, should be able to be played and the video’s host page should be available as well.

Imagine a user adds an inappropriate video to a group due to a mistake or just bad behaviour. The rest of group users should be able to open a poll to remove the video from the group. Each group member should be able to vote a video as inappropriate or appropriate. When a majority is reached, the system should remove or qualify the video as appropriate (not allowing more polls).

Last videos and last polls should be available for each user to facilitate the sharing process. Other statistics related to the videos, users or groups should be available as well in order to promote the interactivity. Depending on which groups the user belongs to, different data should be shown.

Administrator should be able to manage videos, polls, searches, users and groups. Moreover, they will be able to configure the system services settings.1

1 Administrator requirements are not going to be part of next steps in the life cycle because our CMS TYPO3 provides us this functionality.
4.2.2 Non functional requirements

Usability/User interface
The system should be easy to use and as much intuitive as possible. The system should provide the results of the user interactions by AJAX, trying to avoid using back/forward browser commands. The interface should be like a desktop application but its complexity should not discourage users.

Add-on characteristics
The system should be used as an add-on to an already created web. Therefore, the content generated by the tool should be the least traumatic as possible for the actual web content.

TYPO3 integration
The system should be a TYPO3 extension which provides the functionality through several plugins. Adding the plugins to the pages should be simple, using flexforms. Each added plugin should be configurable affording the possibility of adding the same plugin several times with different configuration.

Video providers
Video searching should be as wide as possible. The system should provide videos from several hosting sites. Giving results from only one host would discourage users to use the tool.

Videos display
The video list results from the searches, remote or local, should be paged. System should display video details taken into account which are the more interesting when you are deciding to watch a video. System may give the possibility to watch some videos at the same time, not forcing users to wait while buffering.
4.3 Use cases diagram

Figure 4.1: Use cases diagram
### 4.4 Use case description

<table>
<thead>
<tr>
<th>Use case name:</th>
<th>Search videos on the web.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Summary:</td>
<td>The user searches videos on the web having the possibility to make some operations with the videos, such as play, view host or save.</td>
</tr>
<tr>
<td>Actors:</td>
<td>User, Google Video Search (secondary).</td>
</tr>
<tr>
<td>Preconditions:</td>
<td>User is registered and belongs at least to one group.</td>
</tr>
</tbody>
</table>
| Basic course of events: | 1. User introduces the text query to search.  
2. The system makes the search and displays the search results.  
4. Extension Point: view_host.  
| Alternative paths: | 2. Search with no text. Alert message showed.  
2. No results found. Display an informative message.  
3. No video selected. Alert message shown.  
4. No video selected. Alert message shown.  
5. No video selected. Alert message shown. |
| Extension Points: | play_video: the option “Play” was selected by the user <extend> Play video.  
view_source: the option “View source” was selected by the user <extend> View host page.  
save_video: the option “Save” was selected by the user <extend> Save video. |
<table>
<thead>
<tr>
<th>Use case name:</th>
<th><strong>Search added videos</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Summary:</strong></td>
<td>User searches videos added previously in a group having the possibility to make some operations with the videos such as play, view host or vote them.</td>
</tr>
<tr>
<td><strong>Actors:</strong></td>
<td>User</td>
</tr>
<tr>
<td><strong>Preconditions:</strong></td>
<td>User is register and belongs at least to one group.</td>
</tr>
</tbody>
</table>
| **Basic course of events:** | 1. User makes a group choice.  
2. User selects a filter.  
3. User introduces the text query to search.  
4. The system makes the search and displays the search results.  
5. Extension Point: play_video.  
| **Alternative paths:** | 3. Search with no text. Alert message showed.  
4. No results found. Display an informative message.  
5. No video selected. Alert message shown.  
6. No video selected. Alert message shown.  
7. No video selected. Alert message shown. |
| **Extension Points:** | play_video: the option “Play” was selected by the user <extend> Play video.  
view_host: the option “View source” was selected by the user <extend> View host page.  
vote_video: the option “Vote” was selected by the user <extend> Vote a poll of a video. |
<table>
<thead>
<tr>
<th>Use case name:</th>
<th><strong>View last added videos</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Summary:</strong></td>
<td>The user views last videos added in the groups he belongs to, having the possibility to make some operations with the videos such as play, view host or vote them.</td>
</tr>
<tr>
<td><strong>Actors:</strong></td>
<td>User</td>
</tr>
<tr>
<td><strong>Preconditions:</strong></td>
<td>User is registered and belongs at least to one group.</td>
</tr>
</tbody>
</table>
| **Basic course of events:** | 1. Display the last videos added in the groups.  
2. Extension Point: play_video  
3. Extension Point: view_host  
| **Alternative paths:** | 1. No results found. Display an informative message.  
2. No video selected. Alert message shown.  
3. No video selected. Alert message shown.  
4. No video selected. Alert message shown.  |
| **Extension Points:** | play_video: the option “Play” was selected by the user <extend> Play video.  
view_host: the option “View source” was selected by the user <extend> View host page.  
vote_video: the option “Vote” was selected by the user <extend> Vote a poll of a video.  |

<table>
<thead>
<tr>
<th>Use case name:</th>
<th><strong>Save video</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Summary:</strong></td>
<td>The user adds a video to a group, sharing it with the other users from the group.</td>
</tr>
<tr>
<td><strong>Actors:</strong></td>
<td>User</td>
</tr>
<tr>
<td><strong>Preconditions:</strong></td>
<td>User is registered and belongs at least to one group. A video is selected.</td>
</tr>
</tbody>
</table>
| **Basic course of events:** | 1. User selects the group to add it.  
2. Add the video to the group.  |
| **Alternative paths:** | 2. The video is already added in the group.  
2. The video was inappropriate and was deleted by a poll.  |
<p>| <strong>Extension Points:</strong> | None |</p>
<table>
<thead>
<tr>
<th><strong>Use case name:</strong></th>
<th><strong>Vote a poll of a video</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Summary:</strong></td>
<td>The user values if a video is inappropriate or not.</td>
</tr>
<tr>
<td><strong>Actors:</strong></td>
<td>User</td>
</tr>
<tr>
<td><strong>Preconditions:</strong></td>
<td>User is registered and belongs at least to one group. A video is selected.</td>
</tr>
</tbody>
</table>
| **Basic course of events:** | 1. User votes a video.  
2. The system process the vote. |
| **Alternative paths:** | 2. If it is the first inappropriate vote, the system begins a poll to determine if it is or not appropriate.  
2. With this vote, the majority of votes win. If inappropriateness wins the video is removed from the group. |
| **Extension Points:** | None |

The other use cases are so simple that explain them in this format is not worthy.
4.5 Construction

4.5.1 High-Level Design

4.5.1.1 Conceptual model

The conceptual model is very intuitive. However, it's adequate to explain that there are some many-to-many associations which cause the appearance of their association class concept. These are the join date for user and group, the vote date for user and poll's video or the video details. The video details association class concept helps the system to identify in which groups a video has been added, since a video could belong to several groups.

![Conceptual Model Diagram](image)

Figure 4.2: Conceptual model diagram

The conceptual model is very intuitive. However, it's adequate to explain that there are some many-to-many associations which cause the appearance of their association class concept. These are the join date for user and group, the vote date for user and poll's video or the video details. The video details association class concept helps the system to identify in which groups a video has been added, since a video could belong to several groups.
4.5.1.2 System sequence diagram:
In order to clearly model some of the main events which will be handled by a system, the following system sequence diagram is used.

Figure 4.3: Search videos on the web sequence diagram

Figure 4.4: Search added videos sequence diagram

Figure 4.5: Save video sequence diagram
The sequence diagrams for the rest of system events are not valuable, so will not be shown.

**4.5.1.3 Contracts**

With the system contracts the post-conditions of a system event are specify. We use the conceptual model to identify the objects created, associations formed and attributes modified.

<table>
<thead>
<tr>
<th>Contract:</th>
<th>remotesearch(text: String, offset: Integer)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Responsibilities:</td>
<td>System request Google videos searcher video feeds and display results to the user.</td>
</tr>
<tr>
<td>Preconditions:</td>
<td>None</td>
</tr>
<tr>
<td>Postconditions:</td>
<td>Search instance is created. An association between the User instance and the Search instance is made. A number of Video Feed instances are created. An association between the Video Feed instances and the Search instance is made.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Contract:</th>
<th>localsearch(text: String, offset: Integer, group: Group, filter: String)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Responsibilities:</td>
<td>System makes a search of all the videos added in the group with the option of filtering text by title or description.</td>
</tr>
<tr>
<td>Preconditions:</td>
<td>Group instance must be selected</td>
</tr>
<tr>
<td>Postconditions:</td>
<td>None</td>
</tr>
<tr>
<td>Contract:</td>
<td>savevideo (video:Video,group:Group)</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------------------------</td>
</tr>
<tr>
<td>Responsibilities:</td>
<td>System adds the video feed selected as a video from the group selected.</td>
</tr>
<tr>
<td>Preconditions:</td>
<td>Group instance must be selected</td>
</tr>
</tbody>
</table>
| Postconditions: | Video instance is created  
Video Details instance is created  
An association between the selected Group instance, the Video Details instance and the Video instance is made |

<table>
<thead>
<tr>
<th>Contract:</th>
<th>vote(sign:Integer)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Responsibilities:</td>
<td>System counts the sign of your vote, deleting the video if there is a majority of inappropriate votes.</td>
</tr>
<tr>
<td>Preconditions:</td>
<td>Video instance must be selected.</td>
</tr>
</tbody>
</table>
| Postconditions: | Vote instance is created and associated (if it does not already exist) with a User and a Poll (it is created and associated with the video and the group if it did not exist before).  
Positive or negative counter attribute of the Poll instance is incremented (if vote does not already exist).  
If a majority of inappropriate votes is reached, the Video Details instance associated will be deleted.  
If is the last Video Details instance associated with the Video instance, this will be deleted as well. However, the Vote instance and the Poll instance will be deleted as well without depending on the poll result. |
4.5.2 Low-Level Design

4.5.2.1 Architecture: Model-View-Controller:

The MVC paradigm was used first to solve problems associated with the graphical interface (GUI) for Smalltalk-80 [34]. Users make requests to the GUI, the system analyzes the command requested, manage the data to give a response and then are displayed it to the user. This solution classifies the tasks that should be performed the problem in three specialized groups.

Model: Domain representation of the information that the application manage. It is very common to use persistent storage mechanism, such as a database to store data\(^1\). The architecture doe not specify the data access layer as it is understood to be encapsulated by the Model.

View: Render the data retrieved from the model in a suitable form for good interaction. Several presentations can exist for a single view.

Controller: Manage the events, typical user’s actions and may cause changes on the model.

In Web applications, this design pattern fix fairly well. A HTTP client, a browser or other system, makes a request to a server which will firstly make operations with the data model and then gives a response to return the data.

Web applications developed with MVC architecture have several advantages [35]:

- Our application’s model could have several views. Layer independence allows multiple views for the same model. Consequently the model is easy to manage since all accesses are made in the same place.
- Writing a new view and controller connection with the actual model, solve new types of client’s requirements.
- We obtain a clear design and it is easy to understand how to control the model’s behaviour. Therefore, it will be easier to implement and maintain the application.
- The applications modularity gains in efficiency. Any of the components could be removed and substituted by a new one. It also facilitates parallel development.
- One component such as the model can grow and older versions of the other components can still be used as long as a common interface is maintained.
- MVC makes the application distributable, altering only the startup method of the application.

As we previously explained, we will use the new MVC framework, supplied by TYPO3, to build up our solution with this architecture. Each extension plugin will be a main controller and all the events treated with AJAX will be managed by another controller.

---

\(^1\) TYPO3 uses MySQL but also supports other databases such as Oracle or Postgree
4.5.2.2 Interactions diagrams

We will use sequence diagrams for the interaction diagrams instead of collaboration ones.

System events will be handled by AJAX so that an AJAX controller will manage the events. Each system main action (“remotesearch”, “localsearch”) should be a plugin. Each plugin will have a controller and its aim will be to render the entire interface to provide the user the buttons needed to interact with the system giving responses with AJAX.

Generic event to generate plugin interface:
Figure 4.8: Plugin controller interaction diagram

Translator: It is a class the framework provides to translate some view text in different languages. Translations are given by the administrator and by tags.
remotesearch (text:String, offset:Integer)

toObjectstoObjects: Method that exports the data as an object containing a list of objects. Need the view of each entry that will be separated from the view of the whole list of objects. This allows the developer to give a different view of each entry of a list of data.

Figure 4.9: Remote Search interaction diagram
localsearch (text: String, offset: Integer, group: Group, filter: String)

Figure 4.10: Local Search interaction diagram
savevideo (video: Video, group: Group)

Figure 4.11: Save video interaction diagram
vote(sign: Integer)

Figure 4.12: Vote video interaction diagram
4.5.2.3 Classes Diagram

Figure 4.13: Classes Diagram
4.5.2.4 ER Diagram

- 1 Group may contain N Videos
  1 Video may be contained in N Groups
  This relationship generates a new table called VIDEO_DETAILS which contains the video keys and the group and other interesting attributes, author, tstamp, inappropriate and view_count.

- 1 User may vote N Polls
  1 Poll may be voted by N Users
  This relationship generates a new table called VOTES which contains the user keys and the poll and the tstamp when the user voted.

- 1 User may belong to N Groups
  1 Group may contain N Users
  This relationship generates a new table called JOIN_DATE which contains the user keys and the group, as well as, the tstamp when the user joined the group.
4.5.3 Implementation

Usually, a deep explanation of implementation is not so relevant if there is a good design and the classes are well defined. However, in this thesis, implementation analysis acquires relevance since the system is developed through another big tool, the CMS. Moreover, it is useful to observe how to implement a TYPO3 extension with the new MVC framework.

The TYPO3 extension created is named as “Video Commune”. As a reference to not so crowded and self managed groups. An extension directory view of the “Video Commune” extension may be helpful to understand the explanations below:

<table>
<thead>
<tr>
<th>File</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>ChangeLog</td>
<td>125</td>
</tr>
<tr>
<td>README.txt</td>
<td>80</td>
</tr>
<tr>
<td>ext_am.conf.php</td>
<td>3.5K</td>
</tr>
<tr>
<td>ext_icon.gif</td>
<td>376</td>
</tr>
<tr>
<td>ext_localconf.php</td>
<td>260</td>
</tr>
<tr>
<td>ext_tables.php</td>
<td>3.9K</td>
</tr>
<tr>
<td>ext_tables.sql</td>
<td>2.9K</td>
</tr>
<tr>
<td>localarg.xml</td>
<td>4.2K</td>
</tr>
<tr>
<td>localarg_db.xml</td>
<td>4.4K</td>
</tr>
<tr>
<td>tca.php</td>
<td>12.5K</td>
</tr>
<tr>
<td>doc/manual.xml</td>
<td>512K</td>
</tr>
<tr>
<td>doc/wizard_form.dat</td>
<td>590</td>
</tr>
<tr>
<td>doc/wizard_form.html</td>
<td>3.8K</td>
</tr>
<tr>
<td>configuration/fixformlocal.xml</td>
<td>1.4K</td>
</tr>
<tr>
<td>configuration/fixformremote.xml</td>
<td>5.2K</td>
</tr>
<tr>
<td>configuration/fixformstats.xml</td>
<td>6.4K</td>
</tr>
<tr>
<td>configuration/setup.txt</td>
<td>2.5K</td>
</tr>
<tr>
<td>controllers/class.tx_vcommune_controllers_ajax.php</td>
<td>13.2K</td>
</tr>
<tr>
<td>controllers/class.tx_vcommune_controllers_local.php</td>
<td>0.9K</td>
</tr>
<tr>
<td>controllers/class.tx_vcommune_controllers_remote.php</td>
<td>1.0K</td>
</tr>
<tr>
<td>controllers/class.tx_vcommune_controllers_stats.php</td>
<td>997</td>
</tr>
<tr>
<td>models/class.tx_vcommune_models_videos.php</td>
<td>32K</td>
</tr>
<tr>
<td>views/class.tx_vcommune_views_ajax.php</td>
<td>4.3K</td>
</tr>
<tr>
<td>views/class.tx_vcommune_views_entry.php</td>
<td>350</td>
</tr>
<tr>
<td>views/class.tx_vcommune_views_list.php</td>
<td>352</td>
</tr>
<tr>
<td>views/class.tx_vcommune_views_stats.php</td>
<td>360</td>
</tr>
<tr>
<td>templates/ajax.php</td>
<td>141</td>
</tr>
<tr>
<td>templates/localist.php</td>
<td>18.1K</td>
</tr>
<tr>
<td>templates/remotelist.php</td>
<td>15.9K</td>
</tr>
<tr>
<td>templates/stats.php</td>
<td>55K</td>
</tr>
</tbody>
</table>

Figure 4.15: Video Commune file directory

TYPO3 provides all the functionality we need to manage groups and users. Our system does not need to implement anything to create users or assign groups to them. The web sites administrators would manage everything related to the users and the groups with the TYPO3 BE interface.

First of all, it is required to define the several extension plugins. In our extension there are three plugins “Remote Search”, “Local Search” and “Stats”. Besides having the statistics, the “Stats” plugin contains the last added videos and last open polls. Each plugin should have a controller assigned. This task is made in the “setup.txt” file and these configuration definitions are made by TypoScript, the declarative language mentioned above in this thesis. As we saw in the previous design, the plugins controller only generates a view and the model is not used. The actions
this plugins need to do when the users interacts, are managed by the AJAX controller. This executes actions calling the model and generating a view afterwards. With the data retrieved from the model class, the AJAX view creates data transformed to JSON format. Then the plugin view receives the JSON data created by the AJAX controller and presents the information. In this process Ext JS library participates to make AJAX calls and to transform JSON data into a good display for the user. The controller responsible for managing the AJAX’s call needs to be defined in the setup file by TypoScript.

These plugins have some parameters which are defined when a plugin is added to a page by a flexform. These flexforms are XML files in which the parameters and forms are declared in order to provide a value to them. One flexform is assigned to each plugin controller.

![Figure 4.16: Video Commune Remote Searcher configuration BE form](image)

Some of the concepts obtained in the design need to be stored in the DB. TYPO3 works with MySQL and the SQL statements are established in the “ext_tables.sql” file. This file contains all the tables from the previous ER diagram.

All the model concepts stored in the DB should be available in some way to the administrator in the TYPO3 BE. In this CMS, there is an element called TCA \(^1\) which provides the DB tables a view in the Back-End. TCA is a global array that defines the editable DB tables, the relationship between them and how the fields in the tables are rendered in the BE forms. The “tca.php” file includes the array declarations mentioned.

![Figure 4.17: Video Commune BE elements list view](image)

\(^1\) Table Configuration Array [36]
Another interesting aspect of the extension to be explained is the multi language feature. In TYPO3 we can have a multi language site [37]. Depending on the defined language we will get different output. When we want a translatable text we define in the view a tag like this “%%%tag%%%”. Then, in an XML file called “locallang.xml”, we define different translations of the tag in each language we choose. Finally as we explain in the design, there is a class called “Translator” which is the one that select the corresponding translation. Furthermore, some text in TCA fields can have several translations which are defined in “locallang_db.xml”.

The resulting extension called “Video Commune” is in the TYPO3 extension repository, available for any TYPO3 user who wants to install and use it.

<table>
<thead>
<tr>
<th>Video Commune (vcomune)</th>
<th>Category</th>
<th>Updated</th>
</tr>
</thead>
<tbody>
<tr>
<td>Video Commune</td>
<td>plugin</td>
<td>31.12.2007</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Version</th>
<th>Manual</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0.2</td>
<td>read online</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Downloads (total / this version)</th>
<th>Rating</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>57 / 21</td>
<td>none</td>
<td>Video Commune doesn't store video files, the idea is to share on-line videos links of the main famous providers youtube,... The extension is being developed with the MVC framework, lib 0.0.24 and exists 1.1.1. PHP5 is required. Developed in typo3 4.1.2</td>
</tr>
</tbody>
</table>

Figure 4.18 Video Commune BE Video element view

Figure 4.19 View of the Video Commune extension in the TYPO3 repository
4.6 Demonstration

To demonstrate the performance\(^1\) of the developed TYPO3 extension, a very simple prototype has been built. The demonstration simulates all the video sharing process by some users. We selected Music and News as the main topics the users would discuss. Page structure is very simple. There is a welcome page to login in and when the users are logged they are allowed to enter in the topic discussion. In the discussion there is another welcome page and then the page where discussion is taking place. Only users who belong to the corresponding groups can enter in the discussion pages.

![Figure 4.20 Demo web site page tree](image1)

Each web site user belongs to several groups related to one of these two topics. The users and groups are stored in the folder denominated “users”. For this demo we have created six groups. Three music groups (“electronic”, “hiphop” and “beatles”) and three news groups (“politics”, “science” and “sports”). Six users have also been created, half for each topic.

![Figure 4.21 Demo Web site. Users and groups](image2)

After installing the “Video Commune” extension from the TER\(^2\) using the Extension Manager, we need to include the static extension template. This is simply to provide the web site template with the extension configuration. Once this is finished we can add the plugins to the pages we want.

---

\(^1\) This demonstration was done with: TYPO3 4.1.2, PHP 5.1.6, PHP5-JSON 1.2.1, lib 0.0.24, Ext JS 1.1 and Firefox

\(^2\) TYPO3 Extension Repository
To include the possibility of adding related videos from the internet video hosts, to the group discussion, we add the “Remote Search” plugin to the discussion page. When we add the plugin we have to configure it, providing the storage folder and some search parameters. It is necessary to create the storage folders for all the videos information before adding the plugins, in order to configure them correctly. The storage folders “music videos” and “news videos” are created.

For the News discussion remote search it will be useful to configure a search by date instead of by relevance.

In the welcome page where the users enter in the topic discussion, we can add the “Video stats” plugin. This way, when a user enters in the site, they could see the last videos and last polls. Therefore, users will find it easier to participate in the sharing process. Moreover, statistics from the groups which the user belongs to, will be available. If the users want to make searches to find added videos, we can add the “Video Commune Local Search” in a new page called “Music Videos” or “News Videos” depending on the topic they are participating in.
As the demo web page is already built, we can explain how the process takes place. While the users are participating in the discussions, they can add the videos they consider interesting. Using the “Video Commune Remote Searcher”, users can retrieve videos from the Google Videos search engine (Figure 4.25). Most host web sites are included in the Google Videos searches. Users can play the videos and view the host page (Figure 4.26) where the video belongs to. When the user decides to add a video, all they need to do is to select a group and press “Save” and confirm the dialog (Figure 4.27).
**Figure 4.25: Remote search results view**

<table>
<thead>
<tr>
<th>Video Title</th>
<th>Duration</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vitalic - Poney Part 1: Directed by Pleix</td>
<td>03:03</td>
<td>Jan 20, 2006</td>
</tr>
<tr>
<td>De fiestecita con Vitalic!!!: Fragmento de parte de la sesión que se pudo disfrutar en dancd teatral sala macumba (Madrid). Esa noche Vitalic compartió cartel con Tiga y Etien Allien</td>
<td>02:50</td>
<td>unknown</td>
</tr>
<tr>
<td>Vitalic - La Rock@Creamfields-Andalucía '06: La Rock de Vitalic en Creamfields-Andalucía '06 (12-08-2006)</td>
<td>01:09</td>
<td>Aug 15, 2006</td>
</tr>
<tr>
<td>Vitalic - Poney Part 2 &amp; Repair Machines: Satsumono's Music Shower presents! Vitalic and his awesome 2 song mix from the album &quot;Ok Cowboy&quot; Terrible video quality, but what matters is the music. Enjoy!</td>
<td>08:57</td>
<td>Nov 3, 2006</td>
</tr>
<tr>
<td>De fiestecita con Vitalic!!!: Fragmento de parte de la sesión que se pudo disfrutar en dancd teatral sala macumba (Madrid). Esa noche Vitalic compartió cartel con Tiga y Etien Allien</td>
<td>00:55</td>
<td>unknown</td>
</tr>
</tbody>
</table>

**Figure 4.26: Play and Host page view**

- Vitalic - Poney Part 1
- VITALIC - My Friend Darío (Different Recordings)
- De fiestecita con Vitalic!!!
- Vitalic - La Rock@Creamfields-Andalucía '06
- Vitalic - Poney Part 2 & Repair Machines
- De fiestecita con Vitalic!!!
In the demo, several videos have been added. This way, we can see how other extension areas work. The “Video Commune Local Search” can be used to search videos that other users have added to our groups. In this searcher, we can filter videos by title or description. To make these searches it is always necessary to provide a group. The interface and features are similar to the remote searcher ones (Figure 4.28). Thus, it is easier for the users to use the system. One new feature is the inappropriate button. Using this button, users can consider a video not appropriate for the group and open a poll to remove it from the group. To facilitate the inappropriate process the system uses three intuitive icons. One for the appropriate videos, one for the non qualified videos and one for the videos still not qualified but with an open poll. We see in this demo that some videos are appropriate or not yet qualified (Figure 4.28) while other videos have a poll open to qualify them (Figure 4.29)
As an example, “user3” enters the site and sees the “Video Commune Stats” plugin, where the user can see the last added videos (Figure 4.29), the last open polls (Figure 4.30), and some stats (Figures 4.31-4.41). All this information is related to the groups which “user3” belongs to.

Figure 4.28: Local Search plugin view. Filtering videos with the word “Eminem”

Figure 4.29: Last added videos
Figure 4.30: Last open polls

Figure 4.31: Most viewed videos in all user groups

Figure 4.32: Most viewed videos in one group
Figure 4.33: Most added videos

Figure 4.34: Most searched words

Figure 4.35: Groups with more videos
Figure 4.36: Groups with more appropriate videos

Figure 4.37: Groups with more inappropriate videos

Figure 4.38: Groups with more open polls
Figure 4.39: Users with more videos added

Figure 4.40: Users with more searches

Figure 4.41: Users with more polls opened
In some of the grids, auto refreshing data was suitable as perhaps in another extension component the data is being updated and the data grid changed. For this demo 60 seconds was the time selected to auto refresh. Shorter times cause interferences in the process, for example in the “Video Commune Local Search” plugin.

We make the same for the News web module. The videos, searches and polls of each part are not mixed because a different folder is used for each topic. Moreover, this facilitates the web administration task. In the TYPO3 Back End module videos, polls and searches of each topic can be seen independently (Figure 4.42-4.43). This feature allows the web administrators to divide the tasks and manage the videos separately.

![Figure 4.42: BE music video’s storage folder screenshot](image-url)
Figure 4.43: BE new video’s storage folder screenshot
5 Conclusions

This thesis reviewed multiple methods and tools to enable small sites to share videos. Some of the solutions investigated were high quality ones, but only some of them fitted with the requirements and limitations of an already built small website.

After defining the system characteristics (configurability, feasible, easy to use and easy to integrate), some decisions were taken. Using a tool based on a CMS, we fulfilled these two criteria: “configurability” and “easy integration”. Moreover, using recordings residing on powerful dedicated video providers, the bandwidth problem, which was characteristic of small sites, was solved. Finally, the known AJAX “technology” made the use of the system easy.

The existing tools did not fulfil with the requirements. Therefore, a new tool for TYPO3 was developed following the functional and non-functional requirements.

The developed tool, called “Video Commune”, fulfilled the requirements established in the investigation. With the TYPO3 extension obtained, we tested that the method chosen was appropriate for the scenario mentioned in this thesis. Furthermore, with this software development, we acquired a wide overview of current web technologies. We learnt how to develop with AJAX and discovered new aspects of this powerful “technology”, such as JSON or new libraries, Ext JS¹. We also approached syndication technologies using a feed parser, e.g. Simplepie. Moreover, all the development with this set of technologies was integrated as a TYPO3 extension. This extension followed an architecture pattern really common to a web application, MVC. A new TYPO3 framework² based on MVC was used to develop the extension.

“Video Commune” extension was a good contribution to the TYPO3 Open Source community. Thus far, there were some simple examples³ to understand how to use the new framework. However, extensions based on new framework did not abound. Therefore, our extension would encourage developers to use the new framework.

¹ First stand-alone version, Ext JS 1.1, appeared on August 2007
² First alpha version of TYPO3 MVC new framework appeared on August 2007
³ Some examples extensions: Apples, Bananas and eFAQ
References

http://searchsoa.techtarget.com/sDefinition/0,,sid26_gci508916,00.html [viewed 07/01/2008]

[2] Wikipedia (-) *Content delivery network*  
http://en.wikipedia.org/wiki/Content_Delivery_Network [viewed 07/01/2008]


http://www.adobe.com/devnet/flash/articles/video_guide_03.html [viewed 07/01/2008]

http://www.adobe.com/devnet/flash/articles/flv_encoding_02.html [viewed 07/01/2008]

http://www.mezzoblue.com/archives/2004/05/19/what_is_rssx/ [viewed 07/01/2008]

http://www.youtube.com/t/terms [viewed 07/01/2008]

http://www.youtube.com/dev [viewed 07/01/2008]

http://en.wikipedia.org/wiki/Atom_%28standard%29 [viewed 07/01/2008]

http://www.google.com/coop/cse/ [viewed 07/01/2008]

[11] Boyle, Jaime (-) *Benefits of Being Able to Update Your Own Site*  
http://www.goarticles.com/cgi-bin/showa.cgi?C=645845 [viewed 07/01/2008]

[12] CPD Services (-) *Benefits of a Content Management System*  
http://www.cpdservices.com/53.0.html [viewed 07/01/2008]


http://www.packtpub.com/open-source-cms-award-previous-winners [viewed 07/01/2008]

http://dev.joomla.org/content/view/1137/80/ [viewed 07/01/2008]

http://api.drupal.org/api/file/developer/topics/oop.html/6 [viewed 07/01/2008]

http://archivemati.ca/2006/01/21/drupal-as-a-mvc-framework/ [viewed 07/01/2008]


http://typo3.org/documentation/document-library/extension-manuals/eim2mvc/current/ [viewed 08/01/2008]

http://typo3.org/extensions/repository/view/so_mvc/0.0.43/ [viewed 08/01/2008]

[22] TYPO3.org (-) *Hi, and welcome to TYPO3!*
http://typo3.org/about/new-to-typo3/ [viewed 08/01/2008]

[23] Mediatech (-) *TypoScript*
http://www.mediatech.net/typoscript.html [viewed 08/01/2008]

[24] Skårhøj, Kasper (-) *TypoScript VS XSLT*
http://typo3.org/1198.0.html [viewed 08/01/2008]


http://adaptivepath.com/ideas/essays/archives/000385.php [viewed 08/01/2008]


[30] Ajax Patterns (-) *Frameworks Matrix*  
http://ajaxpatterns.org/Frameworks_Matrix [viewed 08/01/2008]

http://wiki.freaks-unidos.net/javascript-libraries [viewed 08/01/2008]

http://typo3.org/documentation/document-library/extension-manuals/extjs/1.1.1/view/toc/0/ [viewed 08/01/2008]


[34] Burbeck, Steve (1992) *Applications Programming in Smalltalk-80™: How to use Model-View-Controller (MVC)*  

[35] Dass, Karan (-) *Model-View-Controller (MVC) Architecture*  
http://www.indiawebdevelopers.com/technology/java/mvcarchitecture.asp [viewed 08/01/2008]

[36] Skårhøj, Kasper (2005) *TYPO3 Core APIs. Chapter4: Table Configuration Array, STCA*  
http://typo3.org/documentation/document-library/core-documentation/doc_core_api/1.2.0/view/4/1/ [viewed 08/01/2008]

[37] Skårhøj, Kasper (-) *Tips & tricks. Multi language sites in TYPO3*  

[38] Romanov, Sergey (2007) *Joomla hello World MVC component Tutorial*  
http://joomlaequipment.com/content/view/47/74/ [viewed 07/01/2008]

[40] Zend (-) *Zend Framework- Zend_Gdata*  

[41] SimplePie (-) *SimplePie Documentation*  
http://simplepie.org/wiki/ [viewed 08/01/2008]

[42] XAJAX (-) *Docs & Tutorials*  
http://xajaxproject.org/docs.php [viewed 08/01/2008]

[43] Resig, John (-) *How jQuery works*  
http://docs.jquery.com/Tutorials:How_jQuery_Works [viewed 08/01/2008]

[44] Lindley, Cody (-) *Live Examples of jQuery*  
http://docs.jquery.com/Tutorials:Live_Examples_of_jQuery [viewed 08/01/2008]

[45] Moeskau, Brian (200/) *Introduction to Ext (1.1)*  
http://extjs.com/learn/Tutorial:Introduction_to_Ext [viewed 08/01/2008]


[47] Zhuang, Fenqiang (2007) *Using Ext grid form dialog to achieve paging list, create, edit, delete, function*  

http://extjs.com/learn/Tutorial:Creating_JSON_Data_in_PHP [viewed 08/01/2008]


http://typo3.org/documentation/document-library/extension-manuals/efaq/0.1.11/view/toc/0/  
[viewed 08/01/2008]

[51] HotScripts.com (-) *PHP- Scripts and Programs*  
http://www.hotscripts.com/PHP/Scripts_and_Programs/index.html [viewed 08/01/2008]
http://www.sk-typo3.de/ExtJS.340.0.html [viewed 08/01/2008]

[53] PHP.net (-) Documentation
http://www.php.net/docs.php [viewed 08/01/2008]