Comparative analysis of collaborative and scalable e-learning methodologies for control theory

Ciro D. Santilli
Tutor: Roseli Lopes

October 30, 2012

Contents

1 Special thanks

special-thanks

I must express my special thanks to the following people:

- Alexandre Marinazzo from the LSI lab and his co-workers, for tips on Django and web frameworks in general.
- the Django community, for powering such a nice web framework for free
- Google, the Stackoverflow community, and so many bloggers, which just make finding answers to your questions faster
- the linux community in general, for providing free of cost all the necessary tools to make websites and programs that improve the world possible

2 Introduction

introduction

The aim of this project is to study current e-learning methodologies, and after deciding which one is more promising, to develop as many features as possible towards the goal on an order of increasing cost/benefit.

The major advantages of e-learning are:

- re utilizing as much as possible learning material by using the computer instead of giving courses or answering questions live. Clearly, once a question is answered, and if many others have the same question and can find the original question with the answer, teachers don't waste time re-explaining it again. This gain is specially significant if a large number of students uses the system and works on the same material.
- material is thought of beforehand and not made on the spur of the moment, having thus a greater change of being close to optimal
- students have more flexibility in organizing their timetables
- no physical displacement is necessary, lowering costs, and allowing for students who are so far that their presence is impossible to take the course.

Of course, e-learning has some shortcomings, mainly:

- it is harder to feel the emotions of the teacher and other students. This reduces the emotional aspect of learning, which makes students learn by seeing their good friends study a lot (rare in most cases), feeling admiration for the teacher, and feeling fear for the teacher.
- if people are not forced to be present, they may feel less obligation to study and learn, since there are more distractions at home than in the class (not always the case before university in Brazil, where people talk too much in high school)
- using computer screens is tiring on the eyes.

and it is yet an open question whether those will overcome the positive points, and if so, how long will it take.

As the author of this work, I believe that it is only a matter of time before this becomes clear, since in my opinion, only implementing few simple software features will suffice to make e-learning possible, even without taking it into consideration the development of electronic technologies such as higher internet speeds, and e-ind readers which is a potentially slower process than software and methodology development.

After the project requirements are defined, the most promising alternatives to reaching as many those requirements as possible will be studied.

After the alternatives have been analysed, I will implement as much as I can of what I conclude to be the most promising alternative and then present what I have been able to implement.

A part of a short control theory course will be created using the selected methodology in order to show the feasibility of the methods used applied to the teaching of control theory.

3 Use cases

use-cases

The goal of this section is to describe in general what the system does from the point of view of the user.

3.1 Use cases by analogy

use-cases-analogy

Maybe the fastest way of describing the system is by comparing it to other systems. Therefore, if you know what is github [5] / googlecode [6] (git [4] / subversion [1] + web interface) and a bugtracker, I would summarize my desired system as a Github/googlecode with a bugtracker that such that:

- each issue in the bugtracker has a localization id field which indicates what point of the material is the issue about
- users can make groups of users

- users can make collections of localization ids
- users can filter (and therefore give greater priority) only to messages which come from a given group for a given collection of ids.
- users can give read/write permission to entire groups.

The advantage of this system is that teachers would be able to focus first on their students, and students on their classmates, thus complying with their obligations with the institution that pays them (or gives them a diploma in the case of students.)

In this way, if the content created is open source and shared by a large amount of teachers/students, there could be sustainable generation of open source material for all.

Furthermore, I suspect that teachers and students would be even inclined to answer questions of unrelated people if they know the answer and find the question important. This would be even more so if there were reputation metrics that measure how knowledgeable a person is.

What happens is that the barrier between obligation and helping for pleasure gets diluted.

Also, the same system could be useful for software enterprises who work with open sourced material.

3.2 Little history that covers major use cases

use-cases-history

In this section I will tell a little history that illustrates what the system does for users and requires from them. This should be a more complete account of what the system should do.

3.2.1 Scenario

se-cases-scenario

Teacher1 is going to give 2 courses this semester, Course1 and Course2. For each course, he will get help from 1 intern, Intern11, and intern21. Each course will be attended by 60 people, a number which Teacher1 is confident

that with the help of each Intern, that he will be able to answer every question from his students this semester, a common classroom configuration nowadays.

3.2.2 Getting the material ready

First off, Teacher1 has to prepare the text material for his students.

There are two kinds of material

- documents (webpages, pdfs, etc)
- lists of documents and other lists (created by other users, and input in the system database)

For each course, he will have to make a list of material that all the students are obliged to read, in order to pass the exam. Therefore he creates two lists of documents, one called ListCourse1, and the other called ListCourse2. He adds metadata to those lists, in form of a short text explaining that each list contains mandatory reading material for students who are taking each course.

Next, Teacher1 does is to search for already existing material that teaches what he wants to teach so that he does not have to rewrite existing material.

Once the teacher has looked for material, he may choose to:

- create new material if proper one does not exist.
- modify material for which he has edit privileges.
- modify material for which he does not have edit privileges.

3.2.3 Creating new material

The teacher may choose to create teaching material himself if he cannot find any suitable one.

In this process, the teacher would create a new document with a given url, create the text using some markup language, and save it on the server.

The important thing here, is that form this point onwards, if someone wants to edit or view the text, this person must have a privilege to do so, and the teacher has all control over who has this permissions.

3.2.4 Modifying existing material with edit privileges

If the teacher already has edit privileges, the changes he makes will automatically appear on the latest version of the file.

Edits should contain at least two metadata: if the change is minor and the edit summary.

When this is done, people who subscribed to changes on that page will get a notification, and based on the metadata they see, may choose to evaluate the changes made.

If the evaluators agree all is fine, but if not, then he may wish to further modify the changes made, and justify it in the edit metadata.

Finally, if authors cannot reach a consensus, it may be necessary to split the page into two, a process which is called forking in software terminology. Care must be taken when doing so, since while it will allow for each teacher to have exactly what he wants, the amount of users for each page will go down, loosing precious co operational value of mutually answering question and of further improvements. It is a difficult and necessary task for the teacher to strike a balance between those opposing two forces.

3.2.5 Modifying existing material without edit privileges

If on the other hand, the teacher wishes to modify material for which he has no edit privileges, then what he can do is to first copy the material, modify if, and then submit his modifications to those users that have the edit privileges, who should be then notified of this.

Before making large modifications, it is wise to ask those with edit rights if it has a change to me merged, and what criteria it needs to follow to actually be merged. This inquiry can save a lot of time, since either the teacher may implement things directly in the way that editor will accept, and secondly may lead the teacher to completely give up on this modification and modify

someone else's work where the modifications have larger chances of being incorporated.

The users with edit rights can then choose to either accept or reject the merge, and if they reject it, they should justify why. Another option, is that they indicate what would be necessary for the changes to be accept, and if they are simple and desirable enough, even make those changes themselves.

In the meantime, while this evaluation process is going on, the teacher should maintain his modified copy already available to students, and if the changes (patch in software terminology) is accepted, then they can be definitely merged, increasing thus co operative forces on the text.

3.2.6 Reader use cases

reader-use-cases

Once the list of documents has been created, the teacher will publish it, and give the link to students, who are now obliged to read and understand that material in order to pass.

There are a few actions that students may wish to take.

make-a-remark

3.2.6.1 Making a remark

A remark could be either a question, or a general suggestion (for which the student does not know exactly how to implement) about a certain part of a text.

To do so, the student must enter the page and section to where his remark is aimed.

If order for this to work, the authors must give ids for rather small parts of the text, typically for every header, theorem, equation or figure. An automatic system could be devised to insert ids on the input file if the author leaves them out.

Also, it must be easy for users to view the id of the point they want to refer to, so that they can copy and paste it, of even click on it and be able to post.

Furthermore, a search system must exist for all remarks made on a certain id of the text, so that students can ensure, via a simple search system that

their doubt has not been answered before.

If their question has already been answered, then things are even better, since this will not use up more teacher time.

-a-precise-change

3.2.6.2 Suggesting a precise change

It is possible that the student knows exactly what change he wants to make to improve the files.

In this case, teacher should encourage the students to do so: that is, to take the source themselves and improve it.

In doing so, a student:

- helps other students
- learns himself
- makes a good impression on the teacher by showing that he is interested
- saves teacher work

It can be argued that no-one knows what students need better than students themselves.

3.3 Further developments

ther-developments

While the fundamental use cases have been listed here, there are still a few points which are less urgent but would be very interesting to see implemented:

3.3.1 Improved metrics for material

The process of searching for new materials is a complex part of any system, which is used by any user. It can consist of a mix of:

- implicit metrics based on personal estimations of which texts and authors are good, based on what the teacher has read or on the opinions of friends whom he trusts.

- algorithms which make metrics on the materials, and automatically select content which Teacher1 is more likely to like, based on his input

all taking into account which words/expressions the user is looking for.

The key of this process is of course to set metrics, which determine which file should appear above the others on the search results.

I will not focus on this automatic search system, which is a very difficult and researched topic, but I do suppose that a system such as that used by Youtube based on tags and upvotes would be initially sufficient.

It would however be interesting to use more complex metrics, and even allow users to explicitly describe their own metrics with a small piece of code.

3.3.2 Metrics for people

etrics-for-people

Besides ranking materials with several metrics, it is also interesting to rank people with metrics.

This is interesting because:

- it could allow for a person search function that would allow people to find popular writers, to follow their material
- it would be an incentive for people to write, since people want to become famous to get better jobs, and just for the sake of it (game mechanics)

4 Requirements

requirements

This section lists more specific goals which will have to be reached in order to achieve the goals exposed in 3.

4.1 Generation and publishing of learning material

The system must give a way for users to generate textual learning material (referred to simply as "material" from now on), in at least one of the two

target formats: pdf or html. Being able to do both formats from a single input format would be very desirable too (but is currently a difficult task).

4.1.1 Browser editing vs. local file editing

Creation and editing of learning material can be done either on a browser (and therefore somewhat closer to a server), or on several editors which work on local files.

The positive aspects of local file editing methods are:

- current web browser editing methods are much less advanced than local text editing methods
- most input formats need to be processed (latex to pdf, or lightweight markup to html + css) before they can be viewed by the adequate viewers (pdf readers, browsers, etc.). More than that, compilation has to be done several times while writing the input file in order to see what is it turning out like. This kind of compilation process, however, is very cumbersome to carry out directly on browsers as of today, and might be costly to carry on a server side.

The downsides of local editing methods are:

- server/local communication is a point which greatly increases the complexity of the project and the learning curve of local methods of development.
- metadata is meant to be used by the sever, and fits better directly inside a database, forcing users to both commit, then reload the commit web interface, and then finally input the metadata.
- binary data such as photos cannot be shared amongst users on the server in order to save server resources and avoid redundancy, also making distribution of data easier, such as is the case of Wikipedia which centralises all binary data on Wikimedia. This limitation seems hard to circumvent to me.

4.1.2 Input and output formats

The two major output formats are pdf and html, but if it is not possible to do both which one would should be chosen?

The advantages of pdf are:

- mandatory for printing
- wide range of typesetting possibilities for books such as numbering and bibliography support due to latex, which are somewhat lacking/experimental in html outputs.

Advantages of html:

- loads faster
- additional content such as comments or indexes can be added on the same page as the main content without any difficulty
- in order to get user feedback, a browser is going to be necessary at some point of any method, so using html encourages users to give feedback
- it is easier to transform a set of texts into a local copy of those documents (meaning, to transform every link in those html files from a file on a server to a local file iff the file is in the list) since html is plain text, while doing so for pdf would me much easier using latex (I am not sure if this is possible modifying the pdf, but usually modifying existing pdfs is a difficult task)

A linked question is which input format should be used, since there are hard to solve conflicts between the specification of a pdf document and an html one.

A viable solution to the problem is to specify the minimum subset of html/latex which must be generated from a single input file, and implement those items, either with a simplified markup (preferable) or with some harder xml specified language (simpler but harder on the user to type and read). Also,

it is clear that there will always be rarer typesetting features (but no structural features) missing from the chosen subset, and it could simply be chosen that latex will be used to implement those, being the only effort to translate those features to something compatible with html, such as images or mathml. I believe that a subset such as that used in the Connexions [2] system is adequate.

4.2 A privilege/role system

The system must include a way for users to give privileges to others over the internet, in similar fashion to those found in Github or in Joomla. The initial creator of a material has all the privileges, and can choose which privilege will be given to other users. Different privilege levels must exist, being the essential privileges: giving or taking read access to specific users, giving or taking write access to specific users, giving or taking and giving others the ability of giving others read write access to a third user. A level based approach could be used at first, such as read only, read-write access, admin level (can give/take read write access, except of the super admin), and super admin (the only thing he cannot do is to remove privileges from himself).

4.3 Versioning

Versioning has the following objectives:

- allowing pages to refer back to specific versions.

  While it is tempting to always refer to the most recent version of a page in order to get the latest information, this is risky because the information referred to on another page can be missing from the original referencing page.

  There is no simple solution to this impasse, so that it is better to simply reference specific versions, and warn users (and maybe later material creators) that newer versions are available.

- if a mistake or vandalism is commited, one can revert back to a better older version.

- it is possible to see who made good and bad contributions, in order to consider changing their privileges.

In order to achieve those goals, it is fundamental to be able to compare versions, and only differences between versions should be shown in this comparison, ideally in a word-by-word basis as in the Linux word-diff command.

4.4 Addition of metadata

The ability to add several metadata to material by material creators and users, possibly restricted by privileges. The fundamental metadata are tags and a title.

4.5 A bugtracker/comments management system

A bugtracker/comments management system such as Bugzilla or the built-in Github bugtracker system which allows for doubts and suggestions to be made on specific points of the learning material, and so that all users can view those problems and help resolve them. This system must also allow for a user to ask for (email) notifications from other chosen users.

There are some features missing from most bug trackers, which work towards allowing teachers and other students to find the most relevant bugs which they are more likely to know the answer and which are more urgent:

- the creation of groups of people and groups of localization ids (either urls, or urls with ids), together with the ability to filter only issues that come from a certain group of people and a certain group of localization ids.
- the ability for users to vote up on not only on issues which they think are important, but also on answers to those issues which they believe to be effective. This can be coupled with importance metrics so that each user can automatically select which issues are more important to him.

- a field in the bugtracker in which bug submitters write the position of the text at which the question or suggestion is located. For this to work, it would be necessary to attribute ids to different parts of documents

5 Existing solutions

existing-solutions

Many existing systems are close to fulfilling those requirements, but I believe none yet can fulfil all of them in a convenient manner. A major goal of this project will be to analyse those existing systems, and determine which one would be more profitable to modify to reach the desired requirements faster.

5.1 Source code management with bugtracker

This combo has already been used for a long time for the development of open source software and has reached great maturity in this domain of application, however, the techniques used for software should carry to text development since both are nothing but big bunches of organized textual information.

There are however some superficial conflicts that still make this alternative cumbersome for the development of text. I generally:

- the editing is local oriented, suffering thus of difficulties of local development such as steeper learning curve and difficulty of sharing binary resources such as images. There is at least one web interface which allows commit that I know of: Github, which is unfortunately closed source, but this could be the beginning of more.
- files cannot be modified on a one-by-one basis, forcing users who want to make merge requests to either pull the whole repository (which may be large, and very large if a repository has a great amount of images), or for the authors to work with one file per repository, which would require creating and pushing to a large number of different repositories.
- merges are line oriented, while the more natural form of text is natural languages modification is word-wise. It is true however that git can

do word-diffs via the –word-diff option, but the mergers are still line oriented, and thus inefficient. This could however be circumvented by tweaking git by adding a word-wise commit method, chosen by the user at the moment of creation of a repository, and which cannot be changed afterwards.

- there is little information about users, often not even a profile page, since these systems are often developed for projects in which each user

I have focused mostly on Github because it seems to be the most complete web interface for git developement, but if this alternative is to persued, it would be necessary to move to another open source git web interface, such as Gitorious. Also in the case of Gitorious for example, the bug tracker is not integrated, so that it would be necessary to choose, tweak and possibly even integrate a separate bug tracker system into it.

5.2 Wikis

Wikis have one major downside: they are not user oriented because:

- there is very little little permission management. All users have read write access.
- all users are supposed to work on and reach a consensus on a single concept and a way to present it. Obviously, different people will have different ways to present material, and conflict will happen. Furthermore, it might be good to have different points of view on a single concept since each one might be more adequate for each person.

5.3 Blogs and content management systems (CMS)

By CMS it is meant systems like Wordpress, Joomla or Drupal. Blogs have been grouped with them since they can be seen as nothing but a simple individual oriented CMS, being that some of them such as Wordpress are starting to become more and more complete CMS systems. Some of those include even some time of visioning system.

Systems that integrate features of CMS with blogs, wikis and bug-trackers are commonly called project management systems.

5.4 Connexions

Connexions (http://cnx.org/). While it shares some aspects of CMS, it has such a great emphasis in pdf/html conversion.

Connexions uses a native input format based on xml which is a little hard to write manually, but which might be transformed into some lightweight mark-up language.

There are however lots of basic features missing from Connexions, such as users inputting comments about material, or the creation of groups.

5.5 Existing e-learning systems

learning-websites

There are already some projects, such as the Sakai Project (`http://www.sakaiproject.org/`) and and Moodle (`http://moodle.org/`), which are essentially content management systems. I do however feel that those are not made to collaboratively develop very high quality texts, since they focus on many other features, such as student knowledge evaluation, and are often more organized in a manner closely related to traditional classrooms.

6 Comparison of alternatives

In order to choose between the possible alternatives, I have looked at the documentation of several existing projects, and talked with people who have experience on this domain, such as Alexandre Martinazzo and his colleges from the LSI lab.

A determining choice for me was that of implementation language and web framework. The framework is nothing but the library which allows for efficient creation of web sites. The types of jobs it can be used for include:

- database language hooks, particularly through an object representation

to the database

- an html template language, which allows to factor out repeated html code
- a standardized approach to certain design patterns, such as model/view, plug-in architecture

I have first looked from this point of view because all existing solutions are based on some framework, and without knowing the framework, I would be unable to modify any existing project.

Therefore, I decided to first select the language and framework I would use, then develop a simple solution to my problem, and after I had learnt the framework, to integrate my solution with an existing solution.

A comparison of languages can be seen in table 1

language	my knowledge	impression from others	outside server
python	medium/advanced	generally good	yes
php	basic	bad organization	no
ruby	none	too implicit	yes

Table 1: language alternatives tab1

I have excluded C# from consideration, because although a common language for web development, I had no experience with it, and it is linked to Microsoft, which does not favour open source development, meaning that at some point, free solutions would become harder and harder to find.

Therefore I was from the beginning inclined towards going with Python, since most people had a good opinion about it, and I had quite some background with it before. Also, running outside a server is important to me to ease framework administration tasks and to serve for other purposes.

I also took in consideration the existing frameworks for those languages, as summarized in table 2

Here, framework popularity was an important factor, since web development is largely based on mutual community help. Needless to say, measuring framework popularity is a hard taks, but a good measure seems to be the

framework	language	github forks
django [3]	python	4k
php	several	1-4k
ruby-on-rails	ruby	16k

Table 2: framework alternatives tab2

number of forks on Gitbub for each framework, since all frameworks are developed on Github.

I therefore decided to go with the language I knew best, Python, and its main framework, Django, since frameworks seemed to have comparable popularities, and the cost of learning a new language would certainly out-weight the differences of framework popularity.

6.1 Existing django based solutions

In the meanwhile, I have also looked for existing solutions to integrate with.

I know that Django apps are designed to be highly pluggable, so looking for popular apps would be a possible course of action.

6.1.1 Trac bugtracker

trac

My main focus were bug trackers, ticked issuing systems and forums. The most promising alternatives I have found were was the Trac bug tracker [7]. The number one bugtracker based on Django. Allows for custom search fields, but does not seem to be very user oriented, and completely lacks the concept of user group, so important to me. It does support a plug-in system, which would allow for a pluggable implementation of user groups.

Unfortuanatelly Trac is a big system, and modifying if may be a difficult task. Therefore, at this point, I have decided to first create a simple issue tracking system from scratch as an app (a Django plugin), supporting user groups, and later try to incorporate it into a larger system.

7 Schedules

7.1 As 18/09/2012

Out of the total 12 weeks left before the final presentation deadline (11/12/2012), of the project, I intend to allocate tasks as follow:

- 1-2 analysis of existing systems.
- 3-10 implementation of as many missing features as possible for the chosen alternative
- 11-12 creation of a control course using the chosen system

7.2 As 30/10/2012

Looking back on the previous schedule, it was very high level since some important decisions were left, to be made, but after week 1, I had decided to dedicate all my time to learning the Django web framwork, until I was experient enough to develop my own solution.

I underestimated the difficulty of learning a framework, and every seemingly simple new task took me a long time to learn. This is how I used the past weeks:

- 1 analysis of existing systems. decided to use Python Django, and develop first a user-groups app.
- 2-4 learnt the framework basics through tutorials and by making a sample app
- 5-6 started implementing my user groups app which I present today.

Now I feel really much more confortable with the Django framework, and feel that I will be able to better estimate the required time for future tasks. I have decided to spend them as follows:

- 7 finish implementing the user groups app.
- 8 search for existing ticket issuing solutions to integrate with.

  decide if it is worth using them of if it would be better to create a simple ticket issuing app myself.
- 9 finish integration with/implementation of ticket issuing.
- 10-12 develop control learning material using the created components and the proposed methodology.

References

svn [1] Apache subversion. http://subversion.apache.org/, 2012.

Co12 [2] Connexions. http://cnx.org/, 2012.

DD12 [3] Django documentation. https://docs.djangoproject.com/en/dev/, 2012.

git [4] Git. http://git-scm.com/, 2012.

Gi12 [5] Github. https://github.com/, 2012.

Go12 [6] Google code, 2012.

trac [7] The trac project. http://trac.edgewall.org/, 2012.